



Sistema de recomendação na área do desporto

RUI MIGUEL DA SILVA DAMIÃO

Outubro de 2018

Sistema de recomendação na área do desporto

Rui Miguel da Silva Damião

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas de Informação e Conhecimento**

Orientador: Professor Ricardo Almeida

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, outubro 2018

Dedicatória

À minha família e namorada.

Resumo

Este projeto está inserido num ramo de inteligência artificial chamado *Machine Learning* e este baseia-se na ideia de que os sistemas possam aprender com informação, identificar padrões e por sua vez tomar decisões com o mínimo de intervenção humana.

Machine Learning é utilizado no dia-a-dia em recomendações *online* de produtos, deteção de fraudes, anúncios em tempo real, reconhecimento de voz e texto, entre outros.

A presente dissertação tem como objetivo documentar todo o processo de implementação de um sistema de recomendação de anúncios em tempo real na área do desporto.

O sistema de recomendação (baseado em conteúdo) com base no perfil de cada utilizador associa os anúncios desportivos que correspondem com a sua procura ou oferta e envia-lhe uma notificação. Os utilizadores têm acesso às características dos anúncios, mas só poderão ver o proprietário do anúncio e entrar em contacto com ele se usufruírem de uma conta *premium*.

Este sistema permite aos utilizadores criarem e visualizarem anúncios desportivos em várias modalidades, assim estes poderão analisar as melhores ofertas ou procuras atualmente no mercado.

O sistema de recomendação é composto por uma solução *web* desenvolvida em ASP.NET MVC e uma solução móvel desenvolvida em React Native e visa promover uma nova abordagem do processo de captação de jogadores e treinadores.

Palavras-chave: Sistema de recomendação, Desporto, ASP.Net MVC, React Native, SignalR, FireBase

Abstract

This project is embedded in an artificial intelligence branch called Machine Learning and this is based on the idea that systems can learn from information, identify patterns and make own decisions with minimal human intervention.

Machine Learning is used daily in online product recommendations, fraud detection, real-time announcements, voice and text recognition, among others.

The present dissertation aims to document the entire process of implementing a real-time announcements recommendation system in sport area.

The recommendation system (content-based) based on the profile of each user associates the sports announcements that match with his search or offer and sends him a notification. Users have access to the features of the announcements but will only be able to see the owner of the announcement and get in touch with him if they have a premium account.

This system allows users to create and view sports announcements of different modalities, so they can analyze the best deals or searches currently on the market.

The recommendation system consists of a web solution developed in ASP.NET MVC and a mobile solution developed in React Native and aims to promote a new approach to the process of capturing players and coaches.

Keywords: Recommendation system, Sport, ASP.Net MVC, React Native, SignalR, FireBase

Agradecimentos

Gostaria de agradecer aos meus pais, irmã e namorada que sempre me apoiaram durante toda a minha vida académica contribuindo para o seu sucesso.

Agradecer também ao prof. Ricardo Almeida pela disponibilidade apresentada e pelos conselhos dados que foram fundamentais.

Agradeço ainda ao DSI.ISEP, em particular ao Eng. Barros de Oliveira e Tiago Chousal pela oportunidade de realização do projeto, pelo apoio e acreditação do seu propósito.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Problema.....	2
1.3	Motivação.....	2
1.4	Objetivos.....	3
1.5	Resultados esperados.....	3
1.6	Estrutura da dissertação	4
2	Contexto e estado da arte	5
2.1	Contexto	5
2.2	Trabalho relacionado	11
2.3	Tecnologias existentes	15
2.3.1	Aplicação <i>web</i>	15
2.3.2	Aplicação móvel	22
3	Análise de valor	29
3.1.1	Modelo de Desenvolvimento de Novos Conceitos.....	29
3.1.2	Proposta de valor	31
3.1.3	Valor percecionado	32
3.1.4	Analitic Hierarchy Process (AHP)	33
3.1.5	Modelo Canvas	41
3.1.6	Análise <i>SWOT</i>	43
4	Análise e implementação da solução	45
4.1	Design	45
4.1.1	Identificação de requisitos funcionais.....	45
4.1.2	Casos de uso.....	47
4.1.3	Diagramas de sequência de sistema	51
4.1.4	Identificação de requisitos não funcionais	56
4.1.5	Metodologia	56
4.1.6	Modelo relacional	57
4.2	Implementação do sistema	59
4.2.1	Arquitetura	59
4.2.2	Autenticação e autorização	62
4.2.3	Classificação	62
4.2.4	Comunicação.....	64
4.2.5	Dados.....	64
4.2.6	Serviços	64
4.2.7	Log	65
4.2.8	Idioma	65
4.2.9	Testes	65

4.3	Implementação da aplicação <i>web</i>	66
4.3.1	Arquitetura	66
4.3.2	Interface	66
4.4	Implementação da aplicação móvel	67
4.4.1	Arquitetura	67
4.4.2	Interface	68
4.4.3	Android.....	68
4.4.4	iOS	69
4.5	Base de dados.....	69
5	Avaliação da solução	71
5.1	Grandezas e Hipóteses	71
5.2	Metodologia de avaliação	71
5.3	Resultados do inquérito	72
5.4	Conclusão	73
6	Conclusão	75
6.1	Principais conclusões	75
6.2	Objetivos alcançados	76
6.3	Limitações e Trabalho futuro	76
	Anexos.....	80

Lista de Figuras

Figura 1 - Arquitetura .NET	16
Figura 2 - Diagrama de uma aplicação AngularJS MVC	18
Figura 3 - Diagrama de uma aplicação MVC convencional.....	19
Figura 4 - Gráfico com a tendência mundial de pesquisas sobre as tecnologias .NET, AngularJS e Laravel.....	22
Figura 5 - Mapa mundial com a tendência de pesquisas sobre as tecnologias .NET, AngularJS e Laravel.....	22
Figura 6 - Gráfico com a % de perguntas por mês das tecnologias React Native, Xamarin, Ionic e angularUI	24
Figura 7 - Arquitetura do fluxo de dados	27
Figura 8 - Gráfico com a tendência mundial sobre as tecnologias React Native e Xamarin.....	28
Figura 9 - Funcionamento do modelo de desenvolvimento de novos conceitos.	29
Figura 10 - Valores para atribuição do nível de importância de cada critério.....	33
Figura 11 - Valores de IR para matrizes quadradas de ordem n.....	34
Figura 12 - Árvore hierárquica para o desenvolvimento <i>web</i>	35
Figura 13 - Comparação de alternativas para a aplicação <i>web</i>	37
Figura 14 - Árvore hierárquica para o desenvolvimento móvel.....	38
Figura 15 - Comparação de alternativas para a aplicação móvel	40
Figura 16 - Diagrama de casos de uso Parte 1	48
Figura 17 - Diagrama de casos de uso Parte 2	49
Figura 18 - Diagrama de casos de uso Parte 3.....	49
Figura 19 - Diagrama de casos de uso Parte 4.....	50
Figura 20 - Diagrama de sequência “Criar conta”	51
Figura 21 - Diagrama de sequência “Iniciar sessão”	52
Figura 22 - Diagrama de sequência de gestão de anúncios, vídeos e pacotes.....	53
Figura 23 - Diagrama de sequência “Comprar pacote <i>premium</i> ”	54
Figura 24 - Diagrama de sequência “Atualizar taxas de câmbio”	55
Figura 25 - Diagrama de sequência “Gerir pacotes <i>premium</i> ”	55
Figura 26 - Modelo relacional de dados.....	58
Figura 27 - Arquitetura Repository Pattern.....	60
Figura 28 - Arquitetura lógica do sistema.....	61

Lista de Tabelas

Tabela 1 - Comparação de pacotes Fieldoo entre utilizadores.....	13
Tabela 2 - Comparação das soluções estudadas.....	14
Tabela 3 - Tabela comparativa dos critérios definidos	34
Tabela 4 - Comparação do critério “tendência” na app <i>web</i>	35
Tabela 5 - Comparação do critério “experiência” na app <i>web</i>	36
Tabela 6 - Comparação do critério “documentação” na app <i>web</i>	36
Tabela 7 - Comparação do critério “comunidade” na app <i>web</i>	36
Tabela 8 - Comparação do critério “tendência” na app móvel	38
Tabela 9 - Comparação do critério “experiência” na app móvel	39
Tabela 10 - Comparação do critério “documentação” na app móvel	39
Tabela 11 - Comparação do critério “comunidade” na app móvel	39
Tabela 12 - Modelo Canvas.....	42
Tabela 13 - Análise <i>SWOT</i>	43
Tabela 14 - Escala de classificação utilizada no inquérito.....	72
Tabela 15 - Resultado do inquérito de satisfação e de usabilidade.....	72

Acrónimos e Símbolos

Lista de Acrónimos

AHP	<i>Analitic Hierarchy Process</i>
AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Programming Interface</i>
APNS	<i>Apple Push Notification Service</i>
AVD	<i>Android Virtual Device</i>
CLR	<i>Common Language Runtime</i>
COM	<i>Component Object Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CRUD	<i>Create Read Update Delete</i>
CSS	<i>Cascading Style Sheets</i>
DCOM	<i>Distributed Component Object Model</i>
DDD	<i>Domain Driven Design</i>
DOM	<i>Document Object Model</i>
DSC	<i>Data Source Controls</i>
EF	<i>Entity Framework</i>
EJB	<i>Enterprise JavaBeans</i>
FCL	<i>Framework Class Library</i>
FCM	<i>FireBase Cloud Messaging</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model View Controller</i>
MVP	<i>Model View Presenter</i>
MVVM	<i>Model View ViewModel</i>

ORM	<i>Object Relational Mapper</i>
PHP	<i>Personal Home Page</i>
REST	<i>Representational State Transfer</i>
RN	<i>React Native</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SPAs	<i>Single Page Applications</i>
SQL	<i>Structured Query Language</i>
SWOT	<i>Strengths, Weaknesses, Opportunities, Threats</i>
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>

1 Introdução

Neste primeiro capítulo é apresentado um enquadramento teórico sobre o projeto a implementar nas áreas de desporto e informática, de forma a apresentar o tema da dissertação.

De seguida, é abordado o problema principal que desencadeou o desenvolvimento do projeto e as motivações pessoais do autor.

Posteriormente, são definidos os objetivos principais a alcançar, de forma a dar uma resposta positiva ao problema.

Por fim, é realizada a análise dos resultados alcançados e apresentada a estrutura do presente documento.

1.1 Enquadramento

Esta dissertação destina-se a documentar todo o desenvolvimento do projeto realizado na unidade curricular Tese / Dissertação / Estágio do Mestrado no ramo de Sistemas de Informação e Conhecimento do departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto (ISEP) no ano letivo 2017/2018.

A oportunidade de desenvolvimento deste projeto surgiu de uma parceria entre o ISEP e um agente desportivo ligado ao mundo do futebol e consiste na implementação de um sistema de recomendação composto por duas soluções (*web* e *móvel*).

Este sistema pretende aproximar as entidades de várias modalidades, entre si. Entende-se por entidade todos os atletas, treinadores, dirigentes e agentes desportivos de cada modalidade.

Para alcançar o maior número possível de entidades no desporto foi necessário implementar um espaço digital onde estas tenham a possibilidade de criar e procurar anúncios desportivos. Com isto, surge novas oportunidades de negócio e ajuda-as a encontrar as melhores ofertas face às suas necessidades.

1.2 Problema

Atualmente, o sector do desporto está entre as dez atividades económicas mais lucrativas da União Europeia [Comité Olímpico de Portugal, 2012] e todos os dias são concretizados negócios que requerem bastante investimento e existem várias partes envolvidas, como exemplo, as transferências de jogadores entre clubes.

O problema abordado nesta dissertação incide na dificuldade em estabelecer contactos diretos entre clubes, jogadores e treinadores. Normalmente, estes recorrem a agentes desportivos para conseguirem o contacto entre eles, ficando dependentes destes. Este problema afeta principalmente os mais jovens devido à sua falta de experiência profissional.

Este problema ocorre em inúmeras modalidades no desporto que contêm várias entidades envolvidas, desde o atleta ao dirigente da equipa (desporto coletivo) ou entidade patrocinadora (desporto individual).

Com uma vista global do problema, percebe-se por exemplo, que num determinado lugar do planeta poderá haver abundância de jogadores ou treinadores com uma determinada característica e noutro lugar a sua escassez.

Esta falta de comunicação e conhecimento limita bastante as oportunidades de carreira para cada entidade envolvida.

1.3 Motivação

O tema deste projeto cativou bastante o autor a nível pessoal, pois insere-se em duas áreas distintas entre si, sendo elas: desporto e informática. Estas, são áreas da sua preferência e com as quais se identifica bastante.

A possibilidade de aprofundar conhecimentos em engenharia de *software*, sistemas de informação, sistemas móveis e em desporto no desenvolvimento de um projeto é bastante aliciante e motivador por permitir ao autor adquirir mais experiência e conhecimento nas áreas referidas.

1.4 Objetivos

O objetivo deste projeto é implementar um sistema de recomendação na área do desporto, que permita criar uma correspondência entre a oferta e a procura de anúncios desportivos com base no perfil de cada utilizador.

Para concretizar o objetivo geral foram definidos os seguintes objetivos específicos:

- Abranger diversas modalidades (consultar Anexo 1) e o maior número de entidades associadas à respetiva modalidade.
- Disponibilizar todas as informações e notificações em tempo útil para os utilizadores.
- Estabelecer a comunicação entre utilizadores.
- Garantir a qualidade de *software* e a proteção de todos os dados recolhidos.
- Potenciar a utilização de tecnologias para melhorar a experiência do utilizador.
- Determinar a similaridade entre os anúncios de oferta e procura.
- Gerar recomendações.

Neste sentido, pretende-se desenvolver um sistema de recomendação de anúncios desportivos baseado nas características dos anúncios e no perfil de cada utilizador.

1.5 Resultados esperados

É esperado que a aplicação *web* seja a primeira a ser implementada juntamente com todos os componentes que irão compor o sistema, de forma a existir um ambiente de produção estável para os utilizadores. Após esta fase, as aplicações móveis serão implementadas para as plataformas Android e iOS e publicadas nas respetivas lojas.

Durante o processo de implementação das recomendações de anúncios desportivos para os utilizadores, é esperado a escolha de um algoritmo de classificação que satisfaça o objetivo pretendido.

No final da dissertação espera-se concluir todos os pontos abordados no subcapítulo anterior (1.4) de forma a garantir a satisfação do cliente, utilizadores e a qualidade do *software* desenvolvido.

1.6 Estrutura da dissertação

Concluídos os pontos anteriores que têm como objetivo contextualizar e dar a perceber ao leitor o objetivo do trabalho realizado, importa perceber a estrutura da dissertação.

No capítulo 2 são apresentados uma contextualização e um estudo do estado da arte, essencial para que se perceba quais as tecnologias mais utilizadas atualmente no desenvolvimento de *software* e quais os *softwares* já existentes no mercado.

O capítulo 3 descreve uma análise de valor para definir, essencialmente, uma proposta de valor, valor percebido pelos clientes, modelos Canvas e análise *SWOT* (*Strengths, Weaknesses, Opportunities, Threats*) do sistema implementado.

No capítulo 4 é realizada a descrição detalhada do *design* da solução, a implementação do sistema, aplicação *web* e móvel, bem como a base de dados onde os dados irão persistir.

O capítulo 5 apresenta a avaliação sobre a experiência realizada, ou seja, é analisada as respostas dos utilizadores ao inquérito de satisfação e usabilidade da solução. Posteriormente, é feita uma interpretação dos resultados obtidos por parte do autor.

Por fim, no capítulo 6 são efetuadas algumas conclusões sobre o projeto implementado. Aqui são descritas as limitações, possíveis melhorias e trabalho futuro no projeto.

2 Contexto e estado da arte

Este capítulo apresenta o contexto no qual este projeto está envolvido, trabalhos relacionados e as tecnologias atuais que poderão ser úteis no desenvolvimento do projeto.

2.1 Contexto

“Os sistemas de recomendação são ferramentas e técnicas de *software* que fornecem sugestões de itens que possam ser úteis para um utilizador. Estas sugestões dizem respeito a vários processos de tomada de decisão.” [Francesco Ricci et al., 2011]

Existem três categorias de sistemas de recomendação, sendo eles (i) baseado em conteúdo, (ii) filtragem colaborativa e (iii) sistemas híbridos [Souza, 2014] [Casaca et al, 2010].

Baseado em conteúdo

Esta técnica foi a primeira a ser explorada e gera recomendações baseadas na análise de perfil do utilizador e em ações que este tenha realizado no passado.

Um dos fatores que distinguem estes sistemas dos restantes é o facto de gerarem recomendações de itens equivalentes, assim a recomendação de um item é feita com base na semelhança das suas características com as do perfil do utilizador [Bootkrajang, 2017].

Um exemplo de um sistema de recomendação que utiliza esta técnica é o FAB, projeto da Universidade de Stanford, que ajuda os utilizadores a filtrar a quantidade de informação disponível na internet. Este Sistema, utiliza também outras técnicas de forma a eliminar possíveis defeitos de cada abordagem utilizada [Balabanovic & Shoham, 1997].

Um problema comum em sistemas baseados em conteúdo é a inexistência de um perfil definido dos novos utilizadores, a menos que o sistema solicite informações.

Filtragem colaborativa

Uma das vantagens de sistemas que utilizam esta técnica é a possibilidade de recomendar diferentes produtos de diferentes categorias, ou seja, se um utilizador gostar de A e B, um outro utilizador que goste de A poderá também gostar de B [Bootkrajang, 2017].

Um exemplo de um algoritmo de filtragem colaborativa é o slopeOne. Este algoritmo é de fácil implementação e manutenção, eficiente no momento das consultas e razoavelmente preciso nas suas previsões. As predições deste algoritmo são calculadas através da comparação de avaliações efetuadas por utilizadores a certos produtos.

Este determina o quanto melhor um produto é em relação a outro através da subtração da classificação média dos dois. Por sua vez, essa diferença pode ser usada para prever a classificação de um desses produtos [Lemire & Maclachlan, 2005].

Existem também outros algoritmos que podem ser usados neste tipo de sistemas como o *K-Nearest Neighbor* (KNN), *Associations Rules* (AR), *Matrix factorization* (MF) e *Deep Learning* que utilizam funções para calcular a similaridade entre dois itens [Jannach et al, 2011].

Sistemas híbridos

Os sistemas híbridos geram recomendações combinando os métodos baseados em conteúdo com os métodos colaborativos para gerar recomendações de maior qualidade.

Apesar de evitar algumas das fragilidades que um só método apresentaria se fosse implementado individualmente, a implementação desta técnica requer um nível de complexidade elevado por exigir um grande domínio das técnicas anteriormente referidas.

De um modo geral, os sistemas de recomendação, através de algoritmos simples poderão aumentar valor ao negócio em questão porque poderão analisar a similaridade entre as preferências e o perfil do utilizador.

A similaridade pode ser calculada através de funções como (i) métrica, (ii) distância euclidiana, (iii) similaridade dos cossenos, (iv) coeficiente de correlação de Pearson e

(v) coeficiente de similaridade de Jaccard [Huang, 2008] [Rodrigues, 2016] [Tan et al, 2006].

Para o cálculo da métrica, a distância rege-se através de quatro condições, sendo elas [Huang, 2008]:

- A distância entre dois pontos deve ser positiva, ou seja, $d(x, y) \geq 0$.
- A distância entre dois pontos é zero, apenas se forem realmente idênticos ($x = y$).
- A distância deve ser simétrica, isto é, a distância de x para y é o mesmo que a distância de y para x.
- A medida deve satisfazer a desigualdade triangular, que é $d(x, y) \leq d(x, y) + d(y, z)$.

A distância euclidiana calcula a distância entre dois pontos x e y através da fórmula apresentada na seguinte equação: $d(x_i, y_i) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, onde x e y representam o objeto candidato e o perfil do utilizador [Huang, 2008]. Quanto mais próximo de zero for o resultado, mais próximos estão x e y.

A similaridade do cosseno corresponde ao cosseno do ângulo formada pelo par de vetores X e Y. O cosseno é calculado através da seguinte equação:

$$\cos sim = \frac{\sum_i^n x_i * y_i}{\sqrt{(\sum_i^n x_i * x_i) * (\sum_i^n y_i * y_i)}}$$

Quanto mais próximo de zero for o ângulo dos vetores x e y, mais semelhantes são.

O coeficiente de correlação de Pearson é calculado através da seguinte fórmula:

$$r(x, y) = \frac{\sum_i^n (r_{x,i} - r_x)(r_{y,i} - r_y)}{\sqrt{\sum_i^n (r_{x,i} - r_x)^2} \sqrt{\sum_j^n (r_{y,j} - r_y)^2}}, \text{ onde:}$$

- x e y representam os utilizadores a serem correlacionados.
- i corresponde ao item.
- $r_{x,i}$ e $r_{y,i}$ são as classificações que cada utilizador atribuiu ao item i.
- r_x e r_y são as médias aritméticas do conjunto de classificações de cada utilizador.

A última técnica mencionada anteriormente, coeficiente de Jaccard [Tan et al, 2006], é uma medida de similaridade entre dois conjuntos calculada através da equação

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}, \text{ onde:}$$

- f_{01} representa o número de pares de objetos que têm uma classe diferente e o mesmo cluster.
- f_{10} representa o número de pares de objetos que têm a mesma classe e um cluster diferente.
- f_{11} representa o número de pares de objetos que têm a mesma classe e o mesmo cluster.

Estas técnicas são utilizadas em grande escala em lojas *online* mundialmente conhecidas como a Amazon¹, Ebay², entre outras.

Este tipo de sistemas de recomendação, normalmente, é constituído por uma aplicação *web* que interage com o utilizador, sugerindo algo que possa ser do seu interesse [Pazzani, 2007].

A abordagem para o desenvolvimento de aplicações *web* consiste na construção de dois componentes, sendo eles cliente e servidor.

No lado do cliente são desenvolvidas páginas através da utilização de linguagens como HTML, CSS e JavaScript que são interpretadas pelo *browser* do utilizador.

A tecnologia AJAX (*Asynchronous JavaScript And XML*), sendo a mais comum do lado do cliente, permite a este efetuar pedidos ao servidor da aplicação com vista a obter novos dados e através deles atualizar a página automaticamente, tornando-a mais dinâmica. Todo este processo fornece uma melhor experiência ao utilizador na utilização da aplicação *web* [Holdener, 2008].

Até 1996 toda a comunicação HTTP era direcionada pelo cliente, ou seja, qualquer atualização na aplicação era realizada pelo utilizador e começaram a surgir novas tecnologias que permitissem que o servidor comunicasse também com o cliente originando assim uma aplicação *web* em "tempo real".

¹ <https://www.amazon.com/> (acedido em 23/11/2017)

² <https://www.ebay.com/> (acedido em 23/11/2017)

A primeira abordagem a este problema começou em 1996 quando foi possível criar e sustentar uma conexão persistente com o navegador da *web* atribuindo-lhe o nome de Push Technology e mais tarde, em 2000, de Comet [Alinone, 2007].

Neste ano surgem diferentes técnicas de implementação como Polling, Long Polling, Frame Streaming, Server-Sent Events entre outras. Em 2007 surge uma nova técnica chamada Websockets em que consiste numa comunicação bidirecional entre o servidor e o cliente [Alinone, 2011].

No servidor é desenvolvida toda a lógica de negócio da aplicação, usando por exemplo linguagens de programação como C#, PHP, PYTHON e JAVA.

Um componente imprescindível em aplicações *web* é a base de dados, esta serve para armazenar todos os dados relacionados com a aplicação. O SQL é a linguagem usada para a gestão e manipulação de uma base de dados relacional. Estas bases de dados surgiram em 1970 quando um investigador matemático da IBM Dr. Edgar F. Codd propôs que os dados fossem representados como tabelas e, tem sido até então, uma metodologia amplamente adotada [Beaulieu, 2009].

Em meados de 1990, surgiu o primeiro sistema de recomendação a ser comercializado, chamado Tapestry, a utilizar algoritmos colaborativos e de conteúdo na filtragem de *emails* recebidos por um conjunto de utilizadores que pertenciam a uma lista de *emails*, tendo como principal objetivo a redução da quantidade de *emails* da caixa de entrada. Assim, os utilizadores criavam regras (filtros) que representavam as suas preferências pessoais [Goldberg et al, 1992].

Em 1994, surgiu um outro sistema de recomendação implementado pelo grupo de investigação GroupLens da Universidade de Minnesota, baseado em filtragem colaborativa. Este sistema utilizava um algoritmo que se baseava nas características semelhantes entre utilizadores para permitir a formação de uma vizinhança, isto é, um conjunto de utilizadores que possuíssem gostos semelhantes entre si.³

Posteriormente, começaram a surgir outros sistemas como P-tango, Techlens e Feers que recomendavam notícias, artigos científicos e filmes para os utilizadores, respetivamente.

³ <https://grouplens.org/> (acedido em 11/05/2018)

Por fim, para o desenvolvimento de qualquer sistema devem ser adotadas regras de engenharia de *software* que permitam estruturá-las e desenvolvê-las para haver uma separação entre os vários componentes que as integram. Estas regras são chamadas de padrões de arquitetura e devem ser o primeiro passo para o desenvolvimento da solução.

Existem vários estilos arquiteturais e os mais comuns são ⁴:

- **Cliente-Servidor:** separa o sistema em duas aplicações, onde o cliente faz pedidos ao servidor.
- **Arquitetura baseada em componentes:** decompõe a aplicação em componentes lógicos ou funcionais que expõem interfaces de comunicação bem definidas. São exemplos desta arquitetura a COM (*Component Object Model*), DCOM (*Distributed Component Object Model*), CORBA (*Common Object Request Broker Architecture*) e EJB (*Enterprise JavaBeans*).
- **Domain Driven Design (DDD):** é orientado a objetos e foca-se na modelação de um domínio de negócio e na definição de objetos de negócio com base em entidades deste. Esta arquitetura defende um modelo de domínio forte e está ligada fortemente ao ORM (*Object Relational Mapper*) que tem como objetivo ocultar o máximo possível de código, permitindo que os seus modelos se preocupem apenas com a forma como eles são representados e como interagem com outros elementos no sistema.
- **Arquitetura por camadas:** reparte as preocupações da aplicação em grupos (camadas). São exemplos desta arquitetura MVC, MVP e MVVM.
- **Message Bus:** requer um sistema de *software* que possa receber e enviar mensagens usando um ou mais canais de comunicação, de modo que as aplicações possam interagir sem precisarem de conhecer os detalhes específicos sobre cada uma.
- **N-Tier / 3-Tier:** separa os componentes como a Arquitetura por camadas, mas com cada componente localizado num computador fisicamente separado.

⁴ <https://msdn.microsoft.com/en-us/library/ee658117.aspx/> (acedido em 10/12/2017)

- **Arquitetura orientada a objetos:** usa uma abordagem baseada na divisão de responsabilidades de uma aplicação ou sistema atribuído-as aos objetos individuais, cada um contendo os dados e o comportamento relevante para o objeto.
- **Arquitetura orientada a serviços (SOA):** refere-se a aplicações que expõem e consomem funcionalidades como um serviço usando contratos e mensagens.

2.2 Trabalho relacionado

Atualmente, já existem alguns sistemas no desporto como o TalentSpy⁵, DreamFootball⁶, Inthemarket e Fieldoo⁷ que têm o intuito de ajudar todos os elementos ligados ao desporto. Estes sistemas permitem, por exemplo, a criação de relatórios de observação, a criação, edição e organização de perfis de jogadores de futebol e *upload* de vídeos de jogadores, de modo ajudar os agentes e treinadores a efetuar uma análise personalizada de jogadores e equipas respetivamente.

De um modo geral, todos estes sistemas têm o objetivo de encontrar jovens talento e posteriormente proceder ao seu recrutamento através de agentes e clubes.

Talent Spy

A aplicação *web* Talent Spy foi desenvolvida pela F3M Information Systems,SA e lançada a sua 1ª versão *online* em 2013. Atualmente, não existe uma aplicação móvel e a aplicação *web* desenvolvida é constituída por 6 idiomas.

Esta tem como objetivo a procura de jovens talento no futebol e como principais funcionalidades (i) a criação de relatórios de observação, (ii) agendamento de tarefas, (iii) criação e avaliação de perfis de jogadores, (iv) registar os jogos realizados, (v) exportação e partilha dos relatórios criados.

Na aplicação existem dois tipos de acesso, a conta Free de acesso gratuito que permite apenas registar um utilizador com acesso ao *software* e a conta Premium/Corporate permite que cada proprietário registe entre 1 e 25 utilizadores com acesso ao *software*,

⁵ <https://www.ftspy.com/> (acedido em 07/10/2017)

⁶ <https://www.dreamfootball.com/> (acedido em 08/10/2017)

⁷ <https://www.fieldoo.com/> (acedido em 09/10/2017)

dependendo do plano que subscreva. Cada utilizador que adira a uma conta Free poderá experimentar, durante 30 dias, todas as funcionalidades da conta Premium.

DreamFootball

A DreamFootball é uma plataforma digital patrocinada pelo Luís Figo, ex-futebolista português, e tem como objetivo a captação de jovens talento no futebol. Esta aplicação está bastante direcionada para as camadas jovens e o seu funcionamento é idêntico ao da rede social Instagram devido a cada jovem jogador poder fazer *upload* dos seus vídeos, possibilidade de seguir e ser seguido por utilizadores recebendo as suas respetivas notificações e no final de cada semana o *top* 10 dos vídeos mais vistos são avaliados e partilhados pelos clubes que sejam parceiros da DreamFootball.

inthemarket

A aplicação móvel inthemarket é uma rede social para a comunidade dos desportistas e tem como funcionalidades principais a partilha de notícias desportivas, criação e pesquisa de perfis de utilizadores. Esta é gratuita e constituída por cinco modalidades (futebol, basquetebol, andebol, voleibol e futsal), três idiomas (português, polaco e inglês) e está direcionada para duas entidades (jogadores e treinadores).

Fieldoo

Fieldoo é uma aplicação *web* direcionada para os atletas, agentes, olheiros e dirigentes do futebol. Cada um cria o seu próprio perfil e poderá procurar outros perfis através da nacionalidade, posição, estado, entre outras características.

A Fieldoo organiza sessões de captação para explorar atletas profissionais ou para que estes assinem um contrato com um clube. Estes eventos são patrocinados pela Fieldoo e contam com a participação de parceiros oficiais da região onde promovem o evento.

Na tabela 1 é possível ver os preços para ter acesso a funcionalidades como troca de mensagens, ver quem visualiza o seu perfil, descontos nas sessões de captação e entre outras.

Tabela 1 - Comparação de pacotes Fieldoo entre utilizadores

Entidade / Pacote	Base	Avançado
Jogador	<ul style="list-style-type: none"> - 2 mensagens /mês. - Perfil promovido na lista de jogadores. - É possível ver quem visualiza o seu perfil. <p>Valor: 15€</p>	<ul style="list-style-type: none"> - 4 mensagens /mês. - Perfil promovido na lista de jogadores e <i>emails</i> semanais. - É possível ver quem visualiza o seu perfil. - 10% de desconto em sessões de captação e em vídeos. <p>Valor: 25€</p>
Agente	<ul style="list-style-type: none"> - 5 mensagem / dia. - Perfil promovido na lista de agentes. - 1 publicação ativa. <p>Valor: 25€</p>	<ul style="list-style-type: none"> - 10 mensagens / dia. - Perfil promovido na lista de agentes e emails semanais. - 5 publicações ativas. <p>Valor: 50€</p>
Dirigente	É necessário efetuar um contacto por <i>email</i> para obtenção de um valor.	

Para o jogador existem três pacotes adicionais de 10€, 30€ e 50€ para poder enviar 5, 20 e 50 mensagens respetivamente.

Para o agente existe um pacote extra no valor de 200€ onde inclui as funcionalidades do pacote Avançado para publicar eventos de captação de atletas e o número de mensagens e publicações ativas são ilimitadas.

Para finalizar são ilustradas, na tabela 2, as principais características de todos os *softwares* descritos até então para facilitar a sua comparação.

Tabela 2 - Comparação das soluções estudadas

Software / Características	TalentSpy	DreamFootball	inthemarket	Fieldoo
Aplicação web	Sim	Sim	Não	Sim
Aplicação móvel	Não	Sim	Sim	Não
Criação e gestão de perfis	Sim	Sim	Sim	Sim
Notificações em tempo útil	Não	Não	Não	Não
Chat	Não	Não	Não	Não
Upload de vídeos	Não	Sim	Não	Não
Suporte ao cliente	Sim	Sim	Sim	Sim
Idiomas	6	1	3	6
Modalidades	1	1	5	1
Gratuito	Sim	Sim	Sim	Não

Apesar de os sistemas mencionados anteriormente não fornecerem qualquer tipo de recomendação e o autor desconhecer sistemas de recomendação de anúncios na área do desporto, existem alguns na área dos recursos humanos que poderão ser muito semelhantes ao sistema a implementar como é o caso do 2HRT e do Harpoon que atuam em Portugal.

O sistema de recomendação 2HRT é um sistema híbrido porque utiliza técnicas de recomendação baseadas em conteúdo e filtragem colaborativa. Este sistema tem como

objetivo facilitar a atribuição de tarefas aos recursos humanos para ajudá-los a responder de uma forma mais eficiente. Este sistema verifica a lista de trabalhadores e de acordo com a tarefa a realizar associa um trabalhador com base no seu perfil [Casaca et al, 2010].

O sistema de recomendação Harpoon é uma plataforma exclusiva de recrutamento, composta por uma aplicação *web* e uma móvel, e tem como objetivo o recrutamento de candidatos com talento, colocando-os em contacto com as empresas de referência no mercado. Através das motivações, perfis e qualificações dos candidatos faz corresponder as melhores ofertas de trabalho existentes. Esta plataforma já conta com empresas como a Danone, L'Oréal, Jerónimo Martins, Sumol, Compal, Nestlé, entre outras [Jornal Económico, 2016].

2.3 Tecnologias existentes

Neste subcapítulo são caracterizadas várias tecnologias existentes para o desenvolvimento do sistema. Após a caracterização das tecnologias abordadas é realizada uma comparação do suporte da comunidade, documentação e da tendência mundial para se perceber quais as tecnologias mais procuradas de momento.

2.3.1 Aplicação *web*

Para o desenvolvimento de aplicações *web* existem inúmeras tecnologias e são destacadas pela comunidade GitHub e Stack Overflow, a maior comunidade de perguntas e respostas do mundo, estas ⁸:

- .NET
- Angular
- Ruby on Rails
- React
- Django
- Laravel

Irá ser feita a caracterização de três das tecnologias mencionadas anteriormente e por fim a seleção de apenas uma para o desenvolvimento da solução *web*.

⁸ <https://hotframeworks.com/> (acedido em 15/11/2017)

Na escolha das tecnologias foi tido em conta a preferência pelas tecnologias já utilizadas pela instituição (DSI-ISEP) e a experiência pessoal do autor.

A tecnologia .NET foi uma das escolhidas devido à isenção de custos com licenças e pelo suporte da comunidade, enquanto que Angular e Laravel por serem gratuitas e mais recentes.

.NET

Esta tecnologia propõe uma plataforma única para o desenvolvimento de aplicações e foi desenvolvida pela Microsoft. A primeira versão (1.0) foi disponibilizada em 2002, desde então o .NET tem evoluído ano após ano e atualmente já se encontra na versão 4.7.

"A tecnologia .NET consiste em dois componentes principais: uma máquina virtual chamada *Common Language Runtime* (CLR) e a *Framework Class Library* (FCL). Em .NET o compilador não gera código máquina, mas sim uma linguagem intermédia (IL) muito semelhante ao código máquina. Ao programar em C# ou VB.NET o código será compilado para a IL" [Mojica, 2003].

Na figura 1 que se segue é possível ver a arquitetura da tecnologia .NET e o respetivo IDE (*Integrated Development Environment*) que a integra.

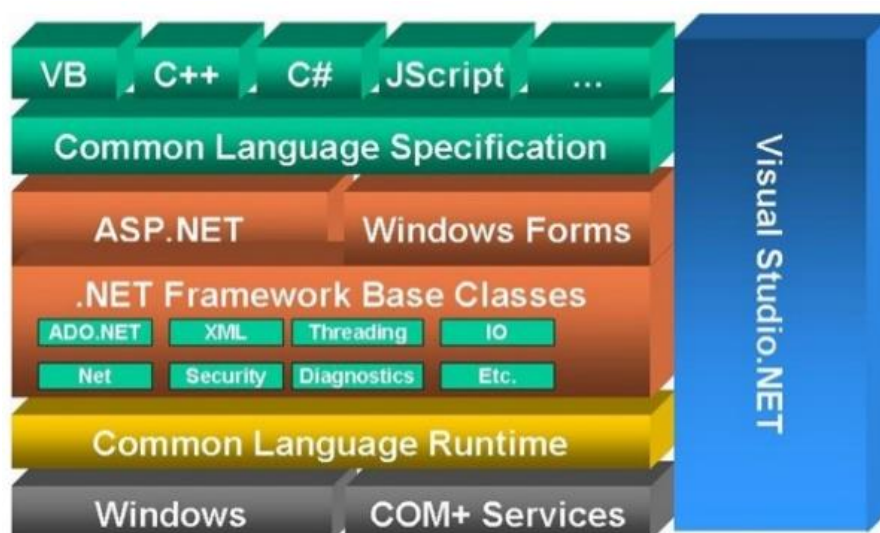


Figura 1 - Arquitetura .NET

(Fonte: *.NET Framework 4.0 – Changes & Benefits*⁹)

⁹ <https://www.slideshare.net/diyaots/net-framework-40-changes-benefits> (acedido em 18/01/2018)

Existem duas tecnologias em .NET direcionadas para o desenvolvimento de aplicações *web*, são elas ASP.NET e ASP.NET MVC. A diferença destas duas tecnologias, como o nome indica, está no uso do padrão arquitetural MVC (*Model-View-Controller*) pela *framework* ASP.NET MVC enquanto que a ASP.NET é baseada em *web forms* e *postbacks*.

Em .NET existem algumas tecnologias para efetuar a comunicação em tempo real entre cliente e servidor, sendo elas:

- Alchemy
- PokelN
- SignalR
- SuperWebSockets

Efetuando uma comparação destas tecnologias [Pimentel, 2014], as que mais se destacam são a PokelN e a SignalR por suportarem mais funcionalidades e plataformas. Em relação à PokelN algumas funcionalidades só estarão disponíveis após a compra da respetiva licença.

Para a leitura e manipulação de dados da base de dados existem duas formas possíveis. O uso direto das classes SqlCommand, SqlDataReader, SqlDataAdapter, e DataSet já existentes que requerem a escrita de código SQL e outras funcionalidades inerentes ou são utilizadas tecnologias que contêm funcionalidades já desenvolvidas, sendo as mais comuns:

- **Entity Framework** (principal ORM da Microsoft)
- **LINQ to SQL**
- **NHibernate**

O Entity Framework é o ORM da Microsoft mais fácil de aprender para quem desenvolve em .NET, sendo o que tem uma maior adesão em relação ao LINQ to SQL e NHibernate¹⁰.

Por fim, a ligação a uma base de dados poderá ser realizada através de¹¹:

- **Data Source Controls (DSC)**: existem vários tipos de DSC como o LinqDataSource, ObjectDataSource e SqlDataSource. A escolha do DSC a utilizar será de acordo com a sintaxe a usar para aceder à base de dados.
- **Providers**: é uma classe .NET que comunica com um tipo específico de base de dados, por exemplo, SQL Server ou Oracle.

¹⁰ [https://msdn.microsoft.com/en-us/library/ms178359\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms178359(v=vs.110).aspx) (acedido em 28/12/2017)

¹¹ [https://msdn.microsoft.com/en-us/library/ms178371\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/ms178371(v=vs.100).aspx) (acedido em 28/12/2017)

- **Conection Strings:** fornece as informações que um *provider* precisa para se conectar a uma base de dados no ficheiro web.config do projeto.

AngularJS

Atualmente, AngularJS é uma tecnologia JavaScript gratuita da Google e foi desenvolvida inicialmente por Miško Hevery e Adam Abrons em 2009. Esta tecnologia utiliza o padrão MVC e é usada, principalmente, no desenvolvimento de SPAs (*Single Page Applications*). Sendo uma tecnologia JavaScript opera do lado do cliente e é uma forma nova e eficiente de desenvolver aplicações *web*.

Nas figuras 2 e 3 que se seguem é possível ver e comparar o diagrama de uma aplicação AngularJS MVC com o diagrama de uma aplicação numa outra tecnologia MVC convencional.

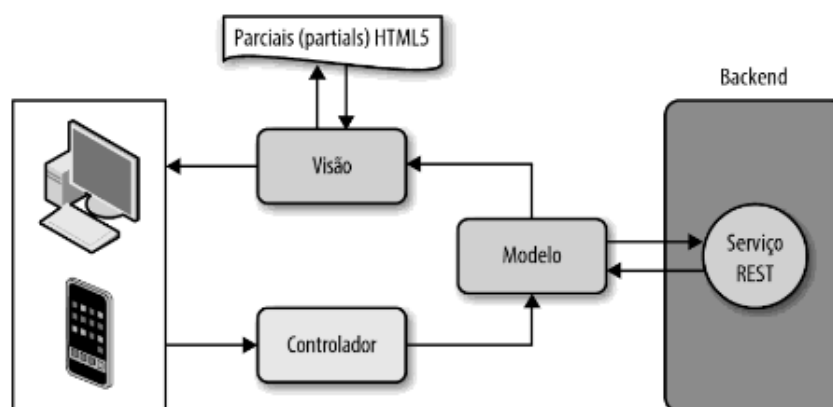


Figura 2 - Diagrama de uma aplicação AngularJS MVC

(Fonte: Williamson, 2015)

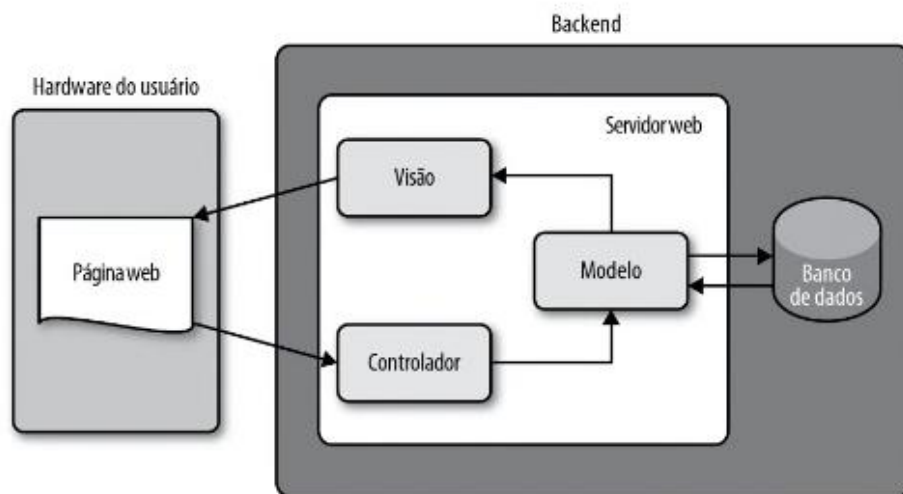


Figura 3 - Diagrama de uma aplicação MVC convencional
(Fonte: Williamson, 2015)

É possível verificar que em aplicações implementadas em AngularJS todos os componentes MVC não estão alojados no servidor (*backend*), mas sim na máquina ou dispositivo do cliente (*frontend*). Este diagrama (figura 2) continua dependente de um serviço externo (REST) onde está toda a lógica e os processos de negócio. Esta dependência tem as suas vantagens, como por exemplo, com a evolução desta tecnologia é normal que algumas aplicações tenham de ser reestruturadas ou reescritas ao contrário do serviço que poderá permanecer igual desde que as tecnologias de *web services* não sofram alterações [Williamson, 2015].

Em AngularJS as tecnologias mais comuns para efetuar a comunicação em “tempo real” entre cliente e servidor são:

- Firebase
- Pusher
- PubNub

Efetuada uma comparação ¹² das três tecnologias mencionadas anteriormente, conclui-se que a mais usada por aplicações *web* é a Firebase e destaca-se a aplicação Twitch que interage diariamente com cerca de 15 milhões de utilizadores.

¹² <https://stackshare.io/stackups/firebase-vs-pubnub-vs-pushers> (acedido em 03/01/2018)

AngularJs possui um *website*¹³ com mais de 2000 módulos desenvolvidos e são de livre acesso. O facto de existirem muitas funcionalidades já implementadas vem facilitar o trabalho de quem desenvolve aplicações nesta tecnologia.

Laravel

Laravel é uma tecnologia PHP gratuita para o desenvolvimento rápido de aplicações *web* e foi desenvolvida por Taylor B. Otwell. Está, atualmente na versão 5.5, utiliza a arquitetura MVC e o seu principal objetivo é ajudar a desenvolver aplicações seguras e com elevado grau de desempenho de forma rápida.

Uma ferramenta utilizada para a criação da interface gráfica chama-se Blade e tem como objetivo a redução e reutilização de código PHP inserido no meio do HTML. Este compilador utiliza uma estrutura de herança e secções como é possível ver no exemplo que se segue.

Página principal:

```
<!-- layout.blade.php -->

<html>
<head>
  <title>App Name - @yield('title')</title>
</head>
<body>
  <div class="container">
    @yield('conteudo')
  </div>
</body>
</html>
```

Conteúdo da página principal:

```
<!-- principal.blade.php -->

@extends('layout')
@section( 'title' , ' Página principal')
@section('conteudo')
  <p>Olá o meu nome é Rui!</p>
@endsection
```

Este exemplo representa uma página com o *layout* definido e outra com o conteúdo da página principal. Assim, é possível perceber melhor o objetivo do Blade, mencionado anteriormente. O termo *@yield* é usado para exibir o conteúdo de uma determinada

¹³ <http://ngmodules.org/> (acedido em 07/01/2018)

secção, o `@section` é responsável em definir uma secção de conteúdo e o `@extends` herda a estrutura de outra página.

Outra ferramenta semelhante ao Blade na tecnologia Laravel é a Twig. Normalmente, esta tecnologia é mais utilizada quando o desenvolvedor da aplicação faz uso de outras tecnologias para além do Laravel. Apesar de não fornecer uma sintaxe rápida e simples como a Blade, adiciona uma nova camada para fornecer mais segurança e funcionalidades adicionais.

Em Laravel as tecnologias mais comuns para efetuar a comunicação em tempo útil entre cliente e servidor são:

- Pusher
- Socket.io
- Firebase

Efetuando uma comparação ¹⁴ das três tecnologias mencionadas anteriormente, conclui-se que a mais usada por aplicações *web* é a FireBase, seguido da Socket.IO e por fim a Pusher.

Para comunicar com a base de dados o Laravel dispõe de um ORM chamado Eloquent ¹⁵ que fornece uma implementação do padrão Active Record. Cada tabela da base de dados tem um modelo correspondente que é usado para interagir com essa tabela. Após o modelo estar criado é possível consultar e inserir dados nas tabelas.

Conclusão

Através do Google Trends foi realizada uma análise da tendência mundial de pesquisas relacionadas com as três tecnologias caracterizadas anteriormente e é possível vê-la na figura 4 que se segue:

¹⁴ <https://stackshare.io/stackups/firebase-vs-pusher-vs-socket-io> (acedido em 28/12/2017)

¹⁵ <https://laravel.com/docs/5.4/eloquent> (acedido em 28/12/2017)

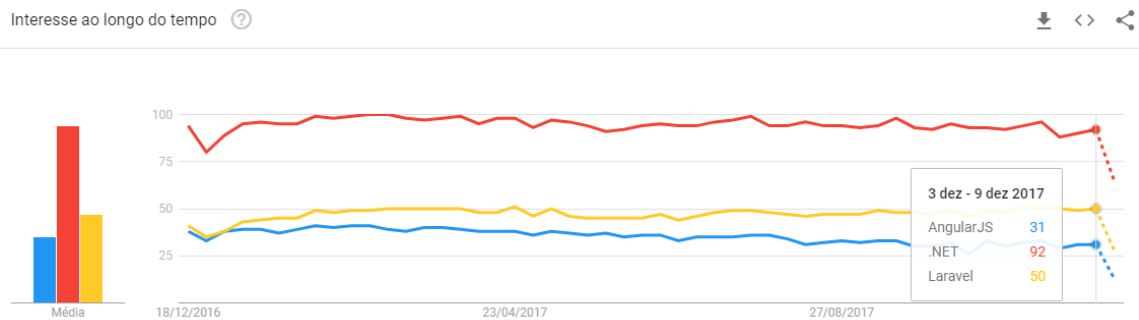


Figura 4 - Gráfico com a tendência mundial de pesquisas sobre as tecnologias .NET, AngularJS e Laravel

(Fonte: Google Trends)

A tecnologia com maior tendência mundial de pesquisas é .NET (vermelho), seguido de Laravel (amarelo) e AngularJS (azul). Para uma melhor percepção destes valores é ilustrado na figura 5 o mapa mundial e a respectiva tecnologia que predomina.



Figura 5 - Mapa mundial com a tendência de pesquisas sobre as tecnologias .NET, AngularJS e Laravel

(Fonte: Google Trends)

É realçado que a tecnologia .NET tem uma comunidade e documentação superior às outras tecnologias devido também ao tempo que está a atuar no mercado.

2.3.2 Aplicação móvel

Para o desenvolvimento de aplicações móveis existem dois tipos de tecnologias, as nativas e as híbridas.

As aplicações nativas são desenvolvidas em linguagens de programação específicas (Objective-C e Java) para um sistema específico (Android, iOS, Windows). Ou seja, se a intenção for disponibilizar uma aplicação para os sistemas mais populares (Android e

iOS) é necessário desenvolver duas aplicações distintas. No caso de ser necessário efetuar alguma alteração implica replicá-la em todos os sistemas.

As aplicações nativas são caracterizadas pela sua rapidez e pelo fácil acesso a funcionalidades do sistema, no entanto exigem maior investimento em desenvolvimento, tanto de tempo como de dinheiro.

Aplicações híbridas são desenvolvidas com tecnologia *web*, como é o HTML5, CSS, JavaScript, entre outras. Estas aplicações têm a vantagem de reutilizarem o mesmo código para os diversos sistemas (Android e iOS) e por sua vez o seu custo é reduzido.

De modo a abranger o maior número de utilizadores face ao tempo para o desenvolvimento, as tecnologias híbridas serão mais vantajosas e por isso a opção escolhida.

As tecnologias híbridas mais utilizadas ou com maior tendência de pesquisas são:

- Ionic
- AngularUI
- React Native
- Xamarin

Irá ser feita a caracterização de duas das tecnologias mencionadas anteriormente e por fim a seleção de apenas uma para o desenvolvimento da solução. A razão pela caracterização de apenas duas tecnologias deve-se ao tempo disponível para a implementação do projeto.

De forma a ajudar na seleção de duas das tecnologias mencionadas anteriormente recorreu-se à ferramenta Stack Overflow Trends para efetuar uma comparação destas tecnologias durante os últimos anos e observar a frequência de perguntas da comunidade como ilustra a figura 6.

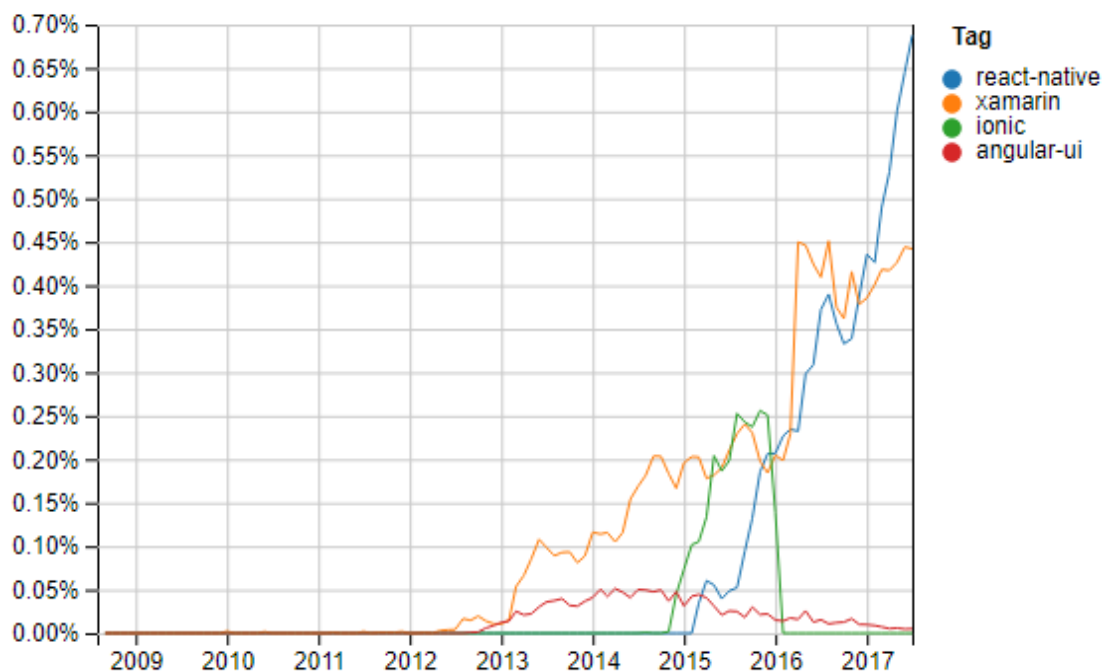


Figura 6 - Gráfico com a % de perguntas por mês das tecnologias React Native, Xamarin, Ionic e angularUI

(Fonte: Stack Overflow Trends)

Com base no gráfico ilustrado na figura 6, conclui-se que as tecnologias Xamarin e React Native têm uma tendência de pesquisa superior em relação às restantes. Assim, irá ser feita a caracterização destas duas tecnologias e por fim a seleção de apenas uma.

Xamarin

Em fevereiro de 2016, a Microsoft anuncia a compra da Xamarin, uma empresa que oferece soluções para o desenvolvimento de aplicações móveis e que permite aos programadores utilizarem a linguagem C# no seu desenvolvimento [Guthrie, 2016].

A Xamarin disponibiliza os produtos Xamarin.Forms, Xamarin.iOS e Xamarin.Android para o desenvolvimento das aplicações nativas. Xamarin.iOS e Xamarin.Android representam as tecnologias da Xamarin que permitem desenvolver aplicações nativas para a plataforma iOS e Android, respetivamente. O produto Xamarin.Forms é uma

tecnologia onde é possível escrever um único código para a interface do utilizador da aplicação e utilizar as funcionalidades de cada plataforma ¹⁶.

A arquitetura de uma aplicação desenvolvida em Xamarin é constituída por 6 camadas ¹⁷, que são:

- **Camada de dados:** onde é realizada a persistência dos dados. Poderá ser uma base de dados SQLite, ficheiros XML ou outro mecanismo adequado.
- **Camada de Acesso aos dados:** está em torno da camada de dados e fornece as operações CRUD (*Create, Read, Update e Delete*) para o acesso e manipulação de dados.
- **Camada de negócio:** contém o modelo e a lógica de negócio da aplicação.
- **Camada de acesso aos serviços:** utilizada para aceder a serviços externos (*web services* e servidores remotos).
- **Camada de aplicação:** possui todo o código que normalmente é específico para uma determinada plataforma e aplicação.
- **Camada de interface do utilizador:** camada que interage com o utilizador.

React Native

React Native é uma tecnologia Javascript desenvolvida pelo Facebook para implementar aplicações nativas para Android e iOS. Estas aplicações podem ser desenvolvidas numa extensão de Javascript e XML, mais conhecida como JSX. Assim, é possível utilizar código Javascript com estruturas parecidas a HTML/XML como é possível ver no código que se segue:

```
var dropdown = <Dropdown> A dropdown list
                <Menu>
                    <MenuItem>Option1</MenuItem>
                    <MenuItem>Option2</MenuItem>
                    <MenuItem>Option3</MenuItem>
                </Menu>
            </Dropdown>;
render(dropdown);
```

¹⁶ https://developer.xamarin.com/guides/cross-platform/getting_started/introduction_to_mobile_development/ (acedido em 27/01/2018)

¹⁷ https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_2_-_architecture/ (acedido em 27/01/2018)

Não é obrigatório o uso de JSX com React Native, mas a sua utilização é bastante recomendada pela simplicidade em definir estruturas em árvore e por sua vez a criação de uma boa estrutura no código.

O React Native usa CSS (*Cascading Style Sheets*) para definir o estilo dos componentes apresentados no ecrã ao utilizador.

Com a utilização desta tecnologia não é necessário efetuar *rebuild* para se ver as alterações efetuadas.

Não é obrigatório a utilização do Android Studio (Android) ou Xcode (iOS) para o desenvolvimento da aplicação, poderá ser utilizado qualquer editor [Eisenman, 2016].

A arquitetura base de uma aplicação desenvolvida em React Native é a seguinte:

- **Android:** diretório onde persiste o código nativo para a plataforma Android. Poderá ser necessário alterar alguns ficheiros dentro do diretório como o gradle e o manifest devido à alteração de permissões ou versão da aplicação para a publicação da mesma na Google Store.
- **iOS:** semelhante ao diretório Android, este diretório persiste o código nativo para a plataforma iOS e raramente sofrerá alterações por parte de quem está a desenvolver.
- **index.js:** é o primeiro ficheiro que aplicação irá executar no arranque da aplicação.

A arquitetura padrão do fluxo de dados em React Native chama-se Flux e está representada na figura 7.

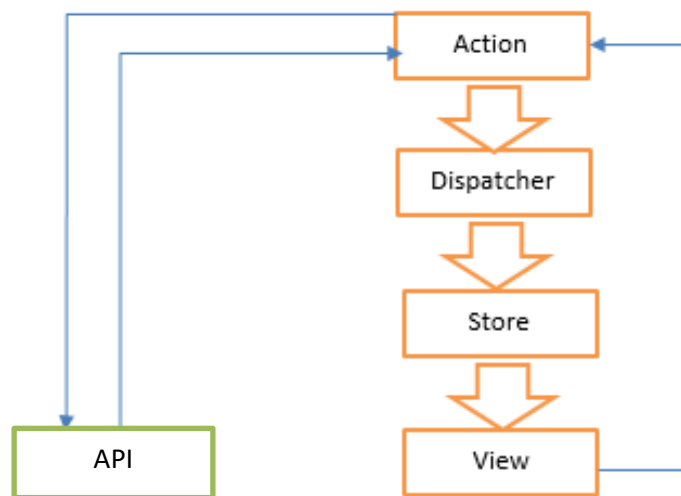


Figura 7 - Arquitetura do fluxo de dados

(Fonte: autor)

A arquitetura Flux é composta por 4 componentes [Fisher & Chen, 2014], sendo eles:

- **Action:** responsável pelos métodos que enviam os dados ao Dispatcher.
- **Dispatcher:** recebe ações e executa-as, modificando o estado e lógica de variáveis ou componentes.
- **Store:** armazena as alterações executadas pelo Dispatcher.
- **View:** desenha a UI de acordo com o estado e lógica de variáveis e componentes.

O 5º componente (API) ilustrado na figura 7 tem um peso importante neste fluxo porque é o componente que fornece toda a informação relacionada com a lógica de negócio da aplicação ao componente Action.

Conclusão

Através do Google Trends foi realizada uma análise da tendência mundial de pesquisas relacionadas com as duas tecnologias caracterizadas anteriormente e é possível observá-la na figura 8 que se segue:

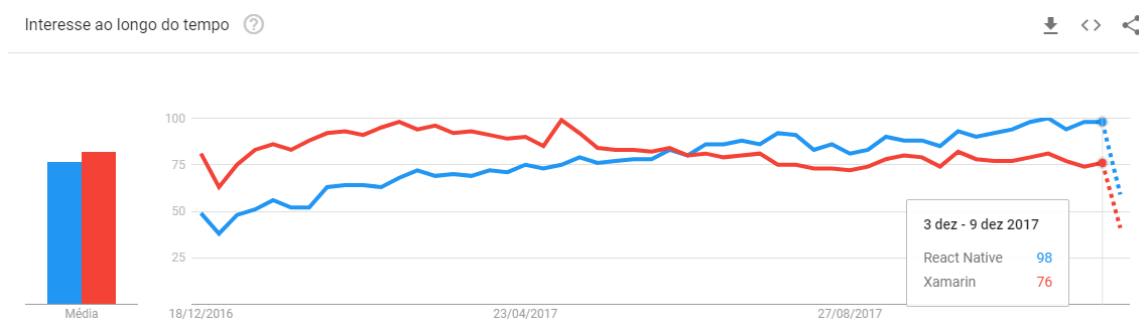


Figura 8 - Gráfico com a tendência mundial sobre as tecnologias React Native e Xamarin

(Fonte: Google Trends)

Apesar do valor da média no último ano ser superior para a tecnologia Xamarin, percebe-se que esta está a perder força e que a recente tecnologia React Native está a ganhar cada vez mais adesão e inclusive ultrapassou a Xamarin a meio do ano de 2017.

O gráfico da figura 8 vem confirmar os dados ilustrados na figura 6 e conclui-se que, atualmente, a tendência pela procura da tecnologia React Native é superior em relação à tecnologia Xamarin.

3 Análise de valor

Neste capítulo é apresentado o conceito de análise de valor, a proposta de valor da solução a implementar, o valor percebido pelo cliente, a utilização do método AHP para a decisão da tecnologia a utilizar, o modelo Canvas e por fim a análise SWOT.

"A análise ou engenharia de valor é um sistema completo para identificar e lidar com os fatores que causam custos ou esforços em produtos, processos ou serviços. Esse sistema usa todas as tecnologias, conhecimentos e habilidades existentes para identificar de uma forma eficiente os custos ou esforços que não contribuem para as necessidades e desejos dos clientes" [Lawrence, 2015].

3.1.1 Modelo de Desenvolvimento de Novos Conceitos

O modelo de desenvolvimento de novos conceitos [Koen et al. 2001] é composto por três componentes chave, representados na figura 9:

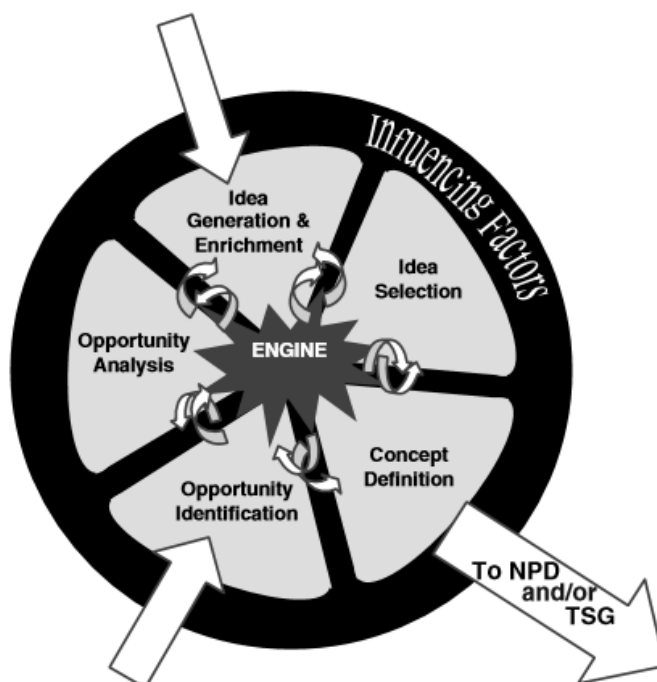


Figura 9 - Funcionamento do modelo de desenvolvimento de novos conceitos.

(Fonte: Koen et al. 2002)

Os componentes representados são:

- **Motor:** representa os níveis superiores de gestão de uma organização que gere os cinco elementos do modelo.
- **Fatores influenciadores:** representam “o quê” ou “quem” influencia o processo de inovação.
- **5 elementos internos:** identificação de oportunidades, a análise de oportunidades, a geração e enriquecimento de ideias, a seleção de ideias e a definição do conceito.

Identificação de oportunidade

A identificação de oportunidades surge quando uma organização identifica oportunidades que possam ser do seu interesse, como por exemplo, uma falha nas tecnologias existentes, um novo processo que reduza os custos, entre outros. Neste projeto, foi identificada uma oportunidade no setor do desporto, onde existe a dificuldade de estabelecer contactos diretos entre jogadores, treinadores, clubes e agentes desportivos.

Análise de oportunidade

Após identificadas as oportunidades, têm de ser analisadas e verificar se realmente são uma mais valia. Este projeto insere-se num mercado mundial, onde qualquer país tem o seu desporto favorito e envolve investimentos constantes. Atualmente, o desporto opta cada vez mais pelo uso de novas tecnologias sendo uma grande oportunidade para o projeto a implementar.

Geração e enriquecimento de ideias

Após ser escolhida uma oportunidade, é necessário gerar ideias que satisfaçam os problemas existentes na mesma. Algumas das ideias para a execução deste projeto surgiram da realização de *brainstormings* com os intervenientes do projeto, onde foi possível identificar, por exemplo, que o processo de correspondência entre os anúncios de oferta e procura poderia ser mais automático e interativo para o utilizador através do uso de notificações.

Seleção de ideia

Com base na geração de ideias realizadas anteriormente é a fase de selecionar a ideia principal. Através das várias ideias que surgiram, verificou-se qual seria a mais atrativa,

constatando-se que a implementação de um sistema de recomendação seria a mais viável e que melhor se enquadrava no problema encontrado.

Definição do conceito

Nesta fase define-se o que é necessário para a solução pretendida, ou seja, definir os objetivos, perceber o mercado atual e identificar as necessidades dos clientes com a finalidade de ter um documento que justifique o investimento necessário.

Após esta fase, o próximo processo é o desenvolvimento efetivo do novo produto, testes, inserção no mercado e avaliação do seu desempenho.

3.1.2 Proposta de valor

“A proposta de valor é definir o conjunto de produtos e serviços que criam valor para um segmento específico de clientes” [Osterwalder, 2011]. Através desta análise, é possível demonstrar a clientes e investidores como esta se diferencia da restante concorrência levando a que estes optem por um determinado produto ou serviço em detrimento de outro.

A solução proposta (sistema de recomendação) promove uma nova abordagem do processo de captação de jogadores e treinadores em várias modalidades desportivas. Isto é possível porque qualquer entidade de uma modalidade pode-se registar e criar anúncios desportivos e posteriormente entrar em contacto com outra entidade para um possível negócio futuro. O facto de a aplicação ter informações e notificações em tempo útil cria uma interação com o utilizador com o objetivo de o alertar para possíveis negócios que poderão ser do seu interesse.

Cada vez mais, os dispositivos móveis são um meio de acesso à informação e estes fornecem-na de forma rápida e simples. A aplicação móvel desenvolvida acrescenta valor para o utilizador porque este poderá ter acesso a informação que lhe é útil em qualquer circunstância do dia.

De forma a ajudar e a manter os utilizadores satisfeitos existe uma secção específica na aplicação *web* e móvel onde estes poderão entrar em contacto com a equipa de apoio e expor todas as suas dúvidas e problemas. Esta interação com os utilizadores poderá estabelecer mais uma vantagem competitiva face à concorrência.

3.1.3 Valor percecionado

“Different customers perceive different value for the same products/services. In addition, organizations involved in the purchasing process can have different perceptions of customers value delivery” [Ulaga & Eggert 2006].

Tal como referido anteriormente, o valor percecionado varia consoante o cliente, sendo normal duas pessoas atribuírem um valor diferente ao mesmo produto ou serviço, o que significa que também é visto de diferentes formas por quem vende e por quem compra.

O valor percecionado pelos clientes desta solução, aparenta ser uma mais valia devido a cada cliente tornar-se mais autónomo e não depender de terceiros para progredir na sua carreira. Este valor poderá variar consoante a idade de cada cliente, sendo superior nos mais jovens devido à sua capacidade de progressão desportiva.

O valor da solução disponibilizada para os desportistas pode ser descrito na forma de benefícios e sacrifícios. Os benefícios chave identificados são:

- **Produto: Solução alternativa:** uma vez que a solução desenvolvida contempla duas aplicações o cliente poderá sempre escolher qual a aplicação a utilizar.
- **Produto: Solução interativa:** existe uma interação entre o sistema desenvolvido e o cliente através de notificações e informações enviadas pelo sistema que poderão ser úteis.
- **Produto: Solução robusta:** o facto de a solução possuir 11 modalidades, 3 idiomas, 12 moedas internacionais e dois tipos de altura (m e ft) tem a capacidade de abranger um maior número de clientes na área do desporto.
- **Serviço: Capacidade de resposta:** tentativa de resposta rápida e um acompanhamento na resolução de problemas dos clientes.

Os sacrifícios identificados são:

- **Produto: Preço associado:** o cliente para possuir o total acesso à informação relacionada com os anúncios desportivos e comunicar com novos utilizadores terá de adquirir um pacote *premium*.
- **Produto: Tempo e esforço:** na primeira utilização da solução, consoante o cliente, é necessário despendar tempo e esforço para completar o perfil, publicar anúncios e conhecer todas as funcionalidades da aplicação.

3.1.4 Analitic Hierarchy Process (AHP)

O método AHP foi desenvolvido por Tomas L. Saaty no início da década de 70 e é o método de multicritério mais utilizado e conhecido no apoio à tomada de decisão em problemas com múltiplos critérios.

Com base nas tecnologias estudadas anteriormente para desenvolvimento *web* (.NET, AngularJS e Laravel) e *móvel* (React Native e Xamarin) foi utilizado o método AHP (Método de Análise Hierárquica) para a definição do grau de importância e o auxílio na tomada de decisão da tecnologia a escolher. Diante destas tecnologias foram definidos quatro critérios: (i) tecnologia mais procurada mundialmente no último ano (tendência), (ii) documentação da tecnologia, (iii) o suporte da comunidade e (iv) a experiência pessoal do autor. O custo de licenças de *software* não foi abordado como critério devido à Instituição estabelecer protocolos e poder usufruir destas tecnologias.

Na figura 10 está representada a escala para classificar o nível de importância de cada critério definido.

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2,4,6,8	Valores intermediários	Quando se procura uma condição de compromisso entre duas definições

Figura 10 - Valores para atribuição do nível de importância de cada critério

(Fonte: Módulo de Análise de Valor, MEI, ISEP 2016)

A figura 11 representa os valores de IR de acordo com número de critérios definidos. Sendo que IR é um índice aleatório calculado para matrizes quadradas de ordem n .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Figura 11 - Valores de IR para matrizes quadradas de ordem n

(Fonte: Módulo de Análise de Valor, MEI, ISEP 2016)

O valor de IR é utilizado para o cálculo da razão de consistência (RC).

Na tabela 3 que se segue é realizada uma comparação entre os critérios definidos:

Tabela 3 - Tabela comparativa dos critérios definidos

Critérios	Tendência	Experiência	Documentação	Comunidade
Tendência	1	8	2	3
Experiência	1/8	1	1/5	1/5
Documentação	1/2	5	1	3
Comunidade	1/3	5	1/3	1

Após a sua normalização, destaca-se da análise dos resultados, o critério “tendência” com um peso de 0.48, que é considerado o mais importante e a “experiência” com um peso de 0.05, o menos importante.

Aplicação web

A figura 12 ilustra a árvore hierárquica para a tomada de decisão de qual alternativa a escolher para o desenvolvimento web.

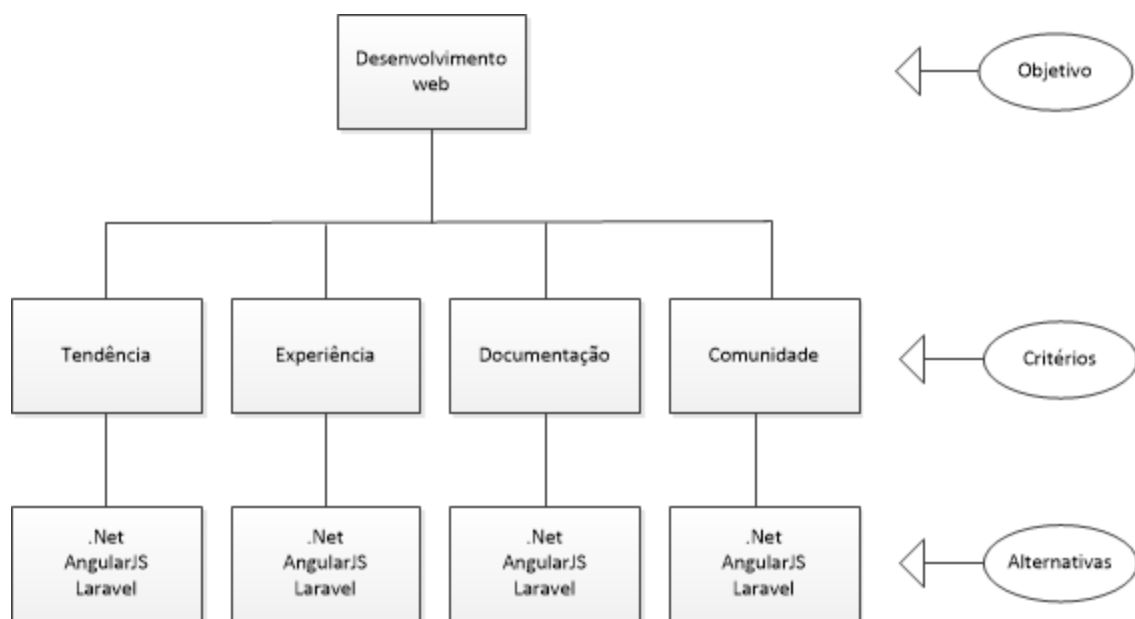


Figura 12 - Árvore hierárquica para o desenvolvimento *web*

(fonte: autor)

Nas tabelas 4, 5, 6 e 7 que se seguem são efetuadas comparações de alternativas face a cada critério definido:

Tabela 4 - Comparação do critério “tendência” na app *web*

Tendência	.Net	AngularJS	Laravel
.Net	1	5	3
AngularJS	1/5	1	1/4
Laravel	1/3	4	1

Após a normalização da tabela 4, destaca-se a superioridade do critério “tendência” para a tecnologia .Net com o valor aproximado de 0.58, seguido de Laravel com 0.26 e AngularJs com 0.1.

Na tabela 5 que se segue, o autor atribui o mesmo valor para as tecnologias .Net e Laravel por ter a mesma experiência em ambas as tecnologias.

Tabela 5 - Comparação do critério “experiência” na app web

Experiência	.Net	AngularJS	Laravel
.Net	1	5	1
AngularJS	1/5	1	1/5
Laravel	1	5	1

Após a normalização da tabela 5, as tecnologias .Net e Laravel possuem o valor de 0.45 e AngularJs o valor 0.1.

Na tabela 6 que se segue, está representada a votação que o autor definiu em relação à documentação de cada tecnologia abordada.

Tabela 6 - Comparação do critério “documentação” na app web

Documentação	.Net	AngularJS	Laravel
.Net	1	5	3
AngularJS	1/5	1	1/2
Laravel	1/3	2	1

Após a normalização da tabela 6, destaca-se a superioridade do critério “documentação” para a tecnologia .Net com o valor aproximado de 0.63, seguido de Laravel com 0.26 e AngularJs com 0.11.

Na tabela 7 que se segue, está representada a votação que o autor definiu em relação à comunidade de cada tecnologia abordada.

Tabela 7 - Comparação do critério “comunidade” na app web

Comunidade	.Net	AngularJS	Laravel
.Net	1	6	3
AngularJS	1/6	1	1/3
Laravel	1/3	3	1

Após a normalização da tabela 7, destaca-se a superioridade do critério “comunidade” para a tecnologia .Net com o valor aproximado de 0.65, seguido de Laravel com 0.25 e AngularJs com 0.1.

Por fim, na figura 13 que se segue é possível ver o resultado da comparação de alternativas para o desenvolvimento de uma aplicação *web*.

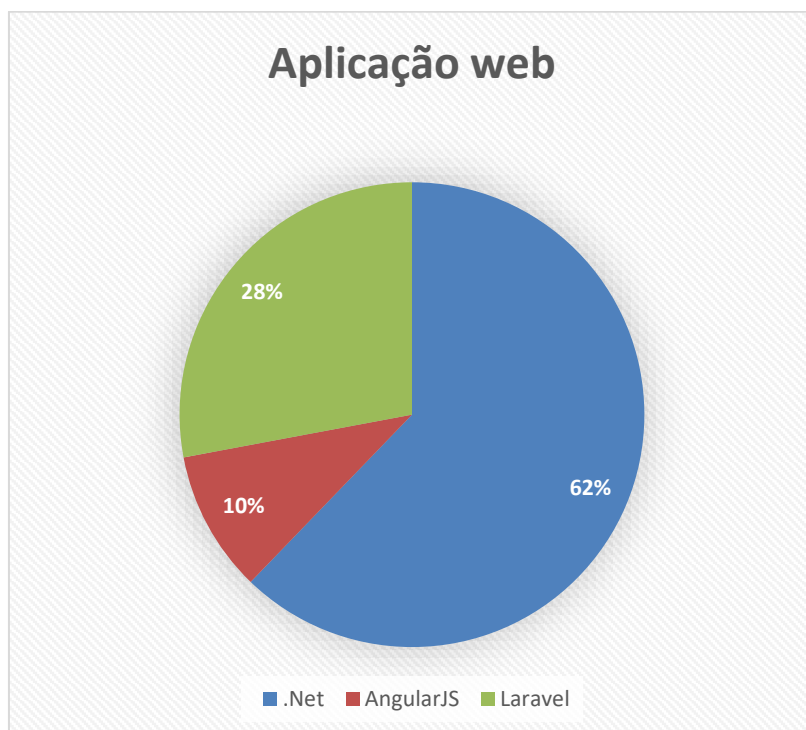


Figura 13 - Comparação de alternativas para a aplicação *web*

(Fonte: autor)

O resultado da comparação de alternativas é superior para a tecnologia .Net com o valor de 62%.

Aplicação móvel

A figura 14 ilustra a árvore hierárquica para a tomada de decisão de qual alternativa a escolher para o desenvolvimento das aplicações móveis.

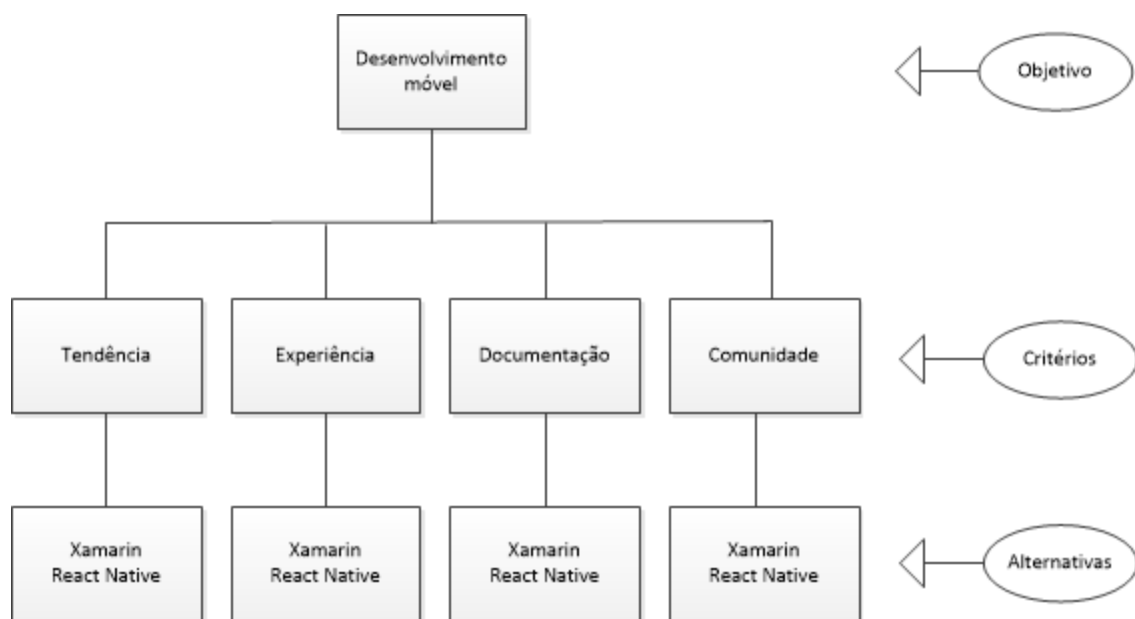


Figura 14 - Árvore hierárquica para o desenvolvimento móvel

(Fonte: autor)

Nas tabelas 8, 9, 10 e 11 que se seguem são efetuadas comparações de alternativas face a cada critério definido:

Tabela 8 - Comparação do critério “tendência” na app móvel

Tendência	Xamarin	React Native
Xamarin	1	1/3
React Native	3	1

Após a normalização da tabela 8, destaca-se a superioridade do critério “tendência” para a tecnologia React Native com o valor de 0.75, seguido de Xamarin com 0.25.

Na tabela 9 que se segue, o autor atribui o mesmo valor para as tecnologias Xamarin e React Native por adquirir a mesma experiência em ambas as tecnologias.

Tabela 9 - Comparação do critério “experiência” na app móvel

Experiência	Xamarin	React Native
Xamarin	1	1
React Native	1	1

Após a normalização da tabela 9, as tecnologias Xamarin e React Native possuem o valor de 0.5 cada uma.

Na tabela 10 que se segue, está representada a votação que o autor definiu em relação à documentação de cada tecnologia abordada. Apesar de ambas as tecnologias estarem igualmente bem documentadas foi atribuído um peso ligeiramente favorável à tecnologia React Native por esta possuir uma documentação mais simples e objetiva para o autor.

Tabela 10 - Comparação do critério “documentação” na app móvel

Documentação	Xamarin	React Native
Xamarin	1	1/2
React Native	2	1

Após a normalização da tabela 10, destaca-se a superioridade do critério “documentação” para a tecnologia React Native com o valor aproximado de 0.67, seguido de Xamarin com 0.33.

Na tabela 11 que se segue, está representada a votação que o autor definiu em relação à comunidade de cada tecnologia abordada. Foi atribuído um peso ligeiramente superior à tecnologia Xamarin por esta possuir uma maior comunidade devido à tecnologia React Native ser mais recente.

Tabela 11 - Comparação do critério “comunidade” na app móvel

Comunidade	Xamarin	React Native
Xamarin	1	3
React Native	1/3	1

Após a normalização da tabela 11, destaca-se a superioridade do critério “comunidade” para a tecnologia Xamarin com o valor de 0.75, seguido de React Native com 0.25. Por fim, na figura 15 que se segue é possível ver o resultado da comparação de alternativas para o desenvolvimento de uma aplicação móvel.

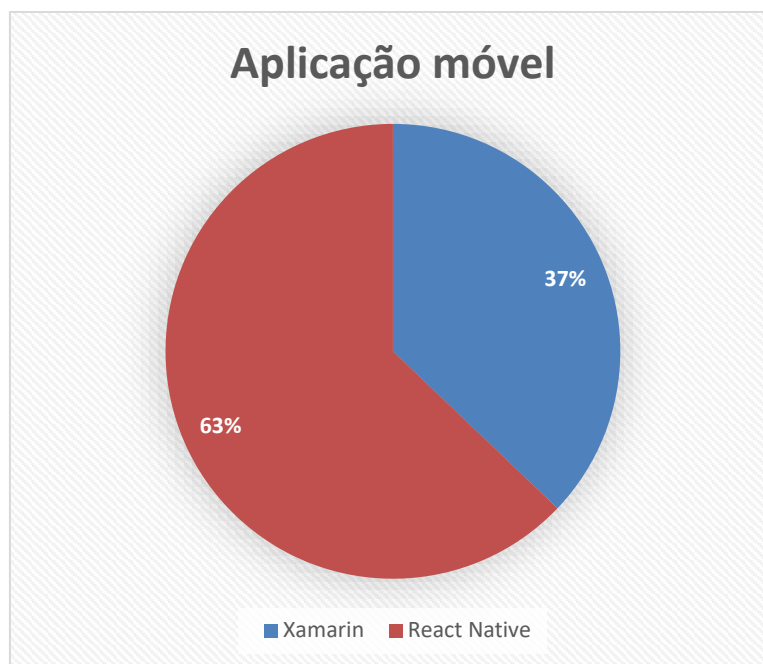


Figura 15 - Comparação de alternativas para a aplicação móvel

(Fonte: autor)

O resultado da comparação de alternativas é superior para a tecnologia React Native com o valor de 63%.

Conclusão

Por fim, é necessário realizar o teste de consistência dos critérios definidos anteriormente para saber se o grau de consistência da matriz de comparação par a par é satisfatório para a escolha da tecnologia *web* e móvel.

Para a determinação do rácio de consistência dos critérios, começa-se por calcular o

vetor próprio $\begin{bmatrix} 1,9890 \\ 0,2042 \\ 1,2946 \\ 0,6786 \end{bmatrix}$, divide-se pelo vetor de pesos dos critérios $\begin{bmatrix} 0,4786 \\ 0,0502 \\ 1,3045 \\ 0,1666 \end{bmatrix}$ e obtém-se

o vetor final $\begin{bmatrix} 4,155 \\ 4,068 \\ 4,251 \\ 4,073 \end{bmatrix}$.

De seguida calcula-se a sua média através da fórmula $\lambda_{\max} = \frac{4.16 + 4.07 + 4.25 + 4.07}{4} \cong 4,14$.

Posteriormente, calcula-se o índice de consistência através da fórmula

$$IC = \frac{\lambda_{\max} - n}{n - 1} = \frac{4.14 - 4}{4 - 1} \cong 0,046.$$

Finalmente, pela aplicação da fórmula $RC = \frac{IC}{IR} = \frac{0,046}{0,9} \cong 0,05$, o resultado é inferior a “0.1”, como tal, consideram-se consistentes as ponderações de critérios efetuadas pelo autor.

3.1.5 Modelo Canvas

De acordo com a tabela representativa do modelo de Canvas (tabela 12) é indicado os parceiros chave que irão ajudar a oferecer a proposta de valor, tais como o Departamento de Sistemas Informáticos (DSI) do Instituto Superior de Engenharia do Porto (ISEP) e um agente desportivo sendo este o detentor do conhecimento na área de negócio em causa.

As atividades chave neste projeto são o desenvolvimento de uma solução *web* e móvel que permitam a criação e gestão de anúncios desportivos, a respetiva correspondência entre a oferta e a procura de anúncios desportivos em várias modalidades, dar suporte ao cliente e estabelecer a comunicação entre utilizadores.

Para o desenvolvimento da solução proposta são identificados estes recursos chave: tecnologias .Net, React Native, Sql Server, acesso à *internet*, emuladores de dispositivos móveis (android e iOS) para o teste das funcionalidades e um alojamento num servidor.

O valor da solução proposta é criar uma correspondência entre a oferta e a procura de anúncios desportivos em várias modalidades, disponibilizar todas as informações associadas em tempo “real” e a possibilidade dos utilizadores comunicarem entre si. Com isto, surgem novas oportunidades de negócio e ajuda todos os utilizadores a encontrar as melhores ofertas face às suas necessidades.

O relacionamento com o consumidor será efetuado através da aplicação *web* e móvel onde haverá uma página para o utilizador expor as suas questões.

Os meios de comunicação que serão estabelecidos com os clientes serão redes sociais, *websites* e *app stores*.

O público alvo está definido na secção de segmentos de clientes, onde são mencionados os atletas, treinadores principais, treinadores adjuntos, preparadores físicos, *personal trainers*, olheiros, agentes e os diretores da direção de clubes.

Na estrutura de custos são mencionados os custos de produção e manutenção da solução, tais como licenças de *software*, tempo despendido no desenvolvimento e testes das aplicações (*web* e *móvel*) e um alojamento num servidor.

As fontes de receitas deste projeto numa fase inicial serão através da aquisição de pacotes *premium* e posteriormente de publicidade.

Tabela 12 - Modelo Canvas

Parcerias Chave	Atividades Chave	Proposta de valor	Relacionamento com o consumidor	Segmentos de consumidores
DSI (ISEP) Agente desportivo/ Ex-futebolista	Correspondência entre a oferta e a procura de anúncios desportivos em várias modalidades	Nova abordagem do processo de captação de jogadores e treinadores em várias modalidades Informações e notificações em “tempo real” Comunicação entre utilizadores	Apoio ao cliente	Atletas Treinadores Treinadores adjunto Preparadores físicos Olheiros Agentes Dirigentes <i>Personal trainers</i>
	Recursos Chave .Net React Native Sql server Emuladores Alojamento num servidor		Canais Redes Sociais <i>Website</i> <i>App stores</i>	

<p>Custos</p> <p>Licenças de <i>software</i></p> <p>Tempo despendido no desenvolvimento e testes das aplicações</p> <p>Alojamento num servidor</p>	<p>Fontes de Receitas</p> <p>Pacotes <i>premium</i></p> <p>Publicidade (futuro)</p>
---	--

3.1.6 Análise *SWOT*

Um bom planeamento e delineação de objetivos é o fundamental para o sucesso de qualquer empresa ou instituição. Para ajudar nesta tarefa é efetuada uma análise *SWOT*, representada na tabela 13, que tem como principal finalidade avaliar o ambiente interno e externo para formular estratégias de negócio para a criação de vantagens competitivas no mercado.

Tabela 13 - Análise *SWOT*

Forças	Fraquezas
<ul style="list-style-type: none"> • Forte conhecimento técnico • Contactos com vários clubes mundiais • Tecnologia utilizada tem provado ser eficiente e estável • Solução abrange várias modalidades desportivas 	<ul style="list-style-type: none"> • Fraca campanha de <i>marketing</i> • Custos fixos com aluguer de servidores
Oportunidades	Ameaças
<ul style="list-style-type: none"> • Aumento do uso de novas tecnologias na área do desporto • Soluções existentes muito específicas 	<ul style="list-style-type: none"> • Regras impostas pela FIFA

A nível interno, foi identificado como força um forte conhecimento técnico na área do desporto, a existência de vários contactos de clubes mundialmente conhecidos o que ajudará na divulgação da solução e as tecnologias utilizadas suportam todos os requisitos funcionais exigidos com um bom desempenho.

Como fraqueza foram identificadas (i) uma fraca campanha de *marketing* devido a promover a solução apenas em redes sociais e através de contactos do agente desportivo que está envolvido no projeto e (ii) os custos fixos mensais de alojamento de um servidor para manter a aplicação *online*.

A nível externo, foi identificado como oportunidade o aumento do uso de novas tecnologias na área do desporto. As soluções existentes no mercado, que foram analisadas, são vocacionadas para uma modalidade específica, e as que abrangem várias modalidades funcionam como uma rede social entre atletas, pelo que se desviam da área de negócio em causa.

Como ameaça foram identificadas algumas regras impostas pela FIFA que poderão limitar o número de utilizadores da solução, uma vez que a idade mínima dos jogadores para transferências internacionais é de 18 anos [Federação Portuguesa de Futebol, 2009].

4 Análise e implementação da solução

O presente capítulo tem como objetivo documentar todo o processo de análise e implementação da solução com base nas tecnologias escolhidas anteriormente.

4.1 Design

A análise de requisitos é vital para o desenvolvimento da aplicação porque é o processo que define as necessidades do cliente e todos os dados recolhidos devem ser detalhados e relevantes para o projeto. Para desenvolver uma boa análise é necessário conhecer bem o problema para avaliá-lo e especificar os requisitos chave e posteriormente efetuar uma revisão.

Após a fase de análise de requisitos, são definidos os casos de uso e diagramas de sequência do sistema para se definir quais as funcionalidades que o sistema irá suportar e perceber todo o seu fluxo.

Por fim, é possível definir um modelo relacional de dados que irá ser a primeira versão da base de dados. Esta poderá sofrer algumas alterações com o decorrer do projeto pela inserção de novas funcionalidades ou alteração de alguma regra de negócio.

4.1.1 Identificação de requisitos funcionais

"Um requisito funcional é a descrição de um comportamento que um sistema irá apresentar sobre condições específicas" [Wiegers & Beatty, 2013].

Os requisitos de *software* definem o funcionamento do sistema e criam um conjunto de regras para que este realize as tarefas necessárias para o bom funcionamento da ideia inicial proposta nesta dissertação.

Após algumas sessões de debate e esclarecimento com todos os intervenientes do projeto foi possível reunir a maior parte da informação que irá envolver o sistema a implementar.

Os atores identificados que irão estar envolvidos neste projeto são:

- Administrador
- Jogador
- Treinador
- Agente
- Dirigente
- Visitante
- Sistema

Seguido da identificação dos atores são descritas as ações chave que irão compor o sistema.

O sistema será composto por 11 modalidades, 3 idiomas (português, inglês e espanhol), 12 moedas internacionais e dois tipos de altura (m e ft) de modo abranger o maior número de utilizadores que possam interagir com o sistema.

Qualquer utilizador, à exceção do visitante, poderá ver, criar, editar, eliminar e filtrar anúncios desportivos na sua respetiva modalidade. Se estes estiverem interessados num anúncio desportivo só poderão ver quem o publicou e interagir através do *chat* da aplicação com essa pessoa se adquirirem um pacote.

Existem vários tipos de pacote tendo eles a duração de uma semana, um mês, três meses e seis meses com os seus respetivos preços e a sua compra será realizada através de uma conta Paypal onde o utilizador poderá escolher o método de pagamento. Numa primeira fase qualquer utilizador poderá adquirir um pacote gratuitamente com a duração de 3 meses.

Sempre que um utilizador iniciar sessão na aplicação será notificado com o número de anúncios que lhe poderão interessar, se estes corresponderem às suas características pessoais (no caso de jogadores e treinadores) ou aos anúncios publicados (no caso de agentes e dirigentes).

O administrador é o responsável pela plataforma, ou seja, para além das funções base de um utilizador normal é adicionado permissões para poder administrar a plataforma, como por exemplo, ver todos os erros gerados pela plataforma, gestão dos pacotes *premium*, gestão de utilizadores, entre outras funcionalidades.

O jogador poderá realizar o *upload* dos seus vídeos a praticar a modalidade para que outros utilizadores o possam observar melhor. O *upload* do vídeo é através do *url* do Youtube para não sobrecarregar o armazenamento no servidor.

O agente e o dirigente após o registo na plataforma terão um tutorial no qual os obriga à criação de um anúncio desportivo. Estes poderão possuir um número ilimitado de anúncios criados e terão acesso a todos os anúncios de oferta de jogadores e treinadores da sua modalidade.

O treinador e o jogador possuem apenas um anúncio cada um e visualizam os anúncios de procura de treinadores e jogadores respetivamente. De modo a mostrar mais informações poderão detalhar o seu histórico como treinador ou jogador.

O visitante vê os últimos anúncios criados de todas as modalidades e poderá registar-se na plataforma através de uma página específica para esse efeito ou através de uma conta Facebook ou Google.

O sistema atualizará as taxas de câmbio de todas as moedas diariamente de forma a manter a coerência dos valores expostos nos anúncios para cada moeda.

Qualquer ação relevante na plataforma, como por exemplo, a criação de um anúncio, o registo de um novo utilizador ou quando dois utilizadores entram em contacto será produzida e destacada uma mensagem em tempo útil na barra do menu onde terá um espaço para as últimas ações realizadas.

4.1.2 Casos de uso

Neste subcapítulo são ilustrados e caracterizados os diagramas de caso de uso da aplicação. Estes servem para descrever as principais funcionalidades do sistema e perceber a interação de cada utilizador com o sistema.

O diagrama de casos de uso será decomposto e caracterizado em quatro partes para que o leitor tenha uma melhor perceção e a sua descrição será representada no formato breve, o formato completo está representado em anexo.

Na figura 16 estão representados todos os atores envolvidos no sistema.

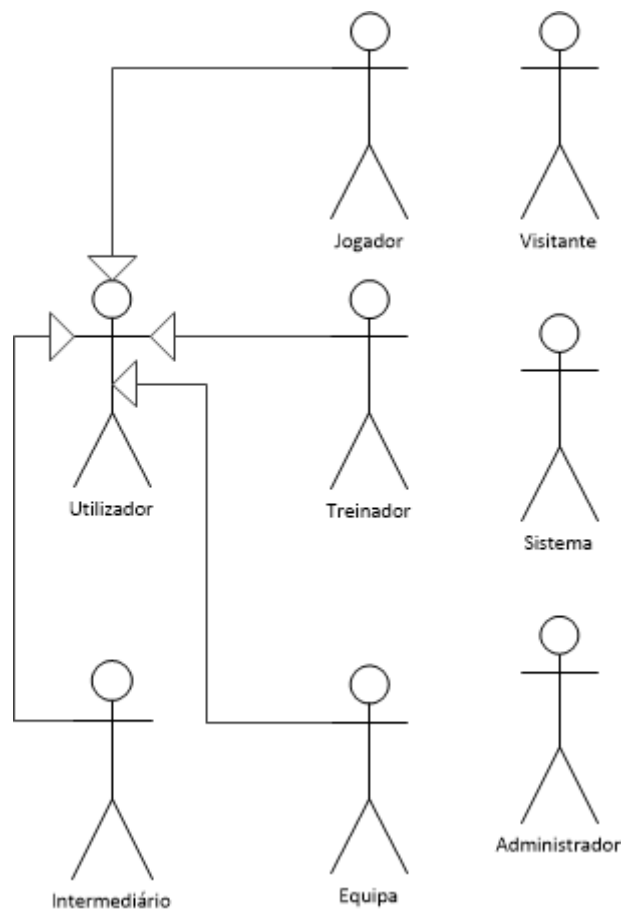


Figura 16 - Diagrama de casos de uso Parte 1

(Fonte: autor)

Os atores envolvidos no sistema são o Utilizador que herda quatro atores (Jogador, Treinador, Equipa e Intermediário), Visitante, Sistema e Administrador.

Na figura 17 estão representadas as principais funcionalidades que qualquer utilizador terá acesso na aplicação.

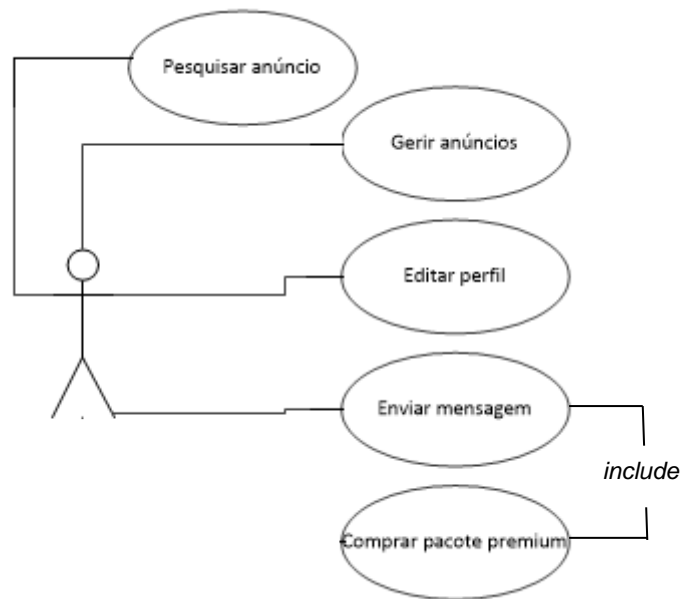


Figura 17 - Diagrama de casos de uso Parte 2

(Fonte: autor)

O utilizador da aplicação *web* e móvel poderá pesquisar anúncios desportivos e posteriormente filtrá-los de acordo com as características que pretender, realizar a gestão (criar, editar, eliminar) dos seus anúncios e perfil, adquirir um pacote *premium*, enviar mensagens a novos utilizadores se tiver adquirido algum pacote e este estiver ativo.

Na funcionalidade “gerir perfil” todos os utilizadores, à exceção do Intermediário, poderão adicionar um histórico profissional ao seu perfil. O ator Jogador tem uma funcionalidade adicional como ilustra a figura 18.

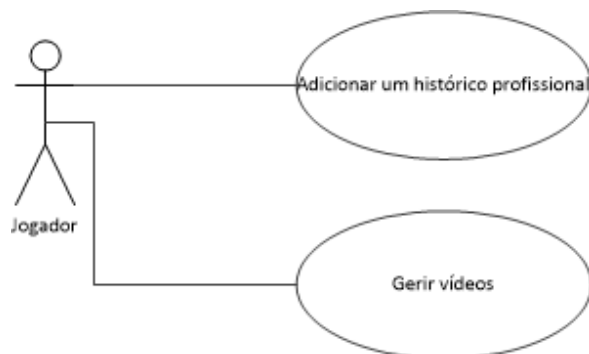


Figura 18 - Diagrama de casos de uso Parte 3

(Fonte: autor)

A funcionalidade “Gerir Vídeos” permite a este, para além da gestão dos vídeos, mostrar a outros utilizadores as suas qualidades em formato de vídeo para que estes possam analisar e avaliar o jogador, assim o jogador dá-se a conhecer de uma melhor forma.

Na figura 19 estão representadas as principais funcionalidades dos atores Visitante, Sistema e Administrador.

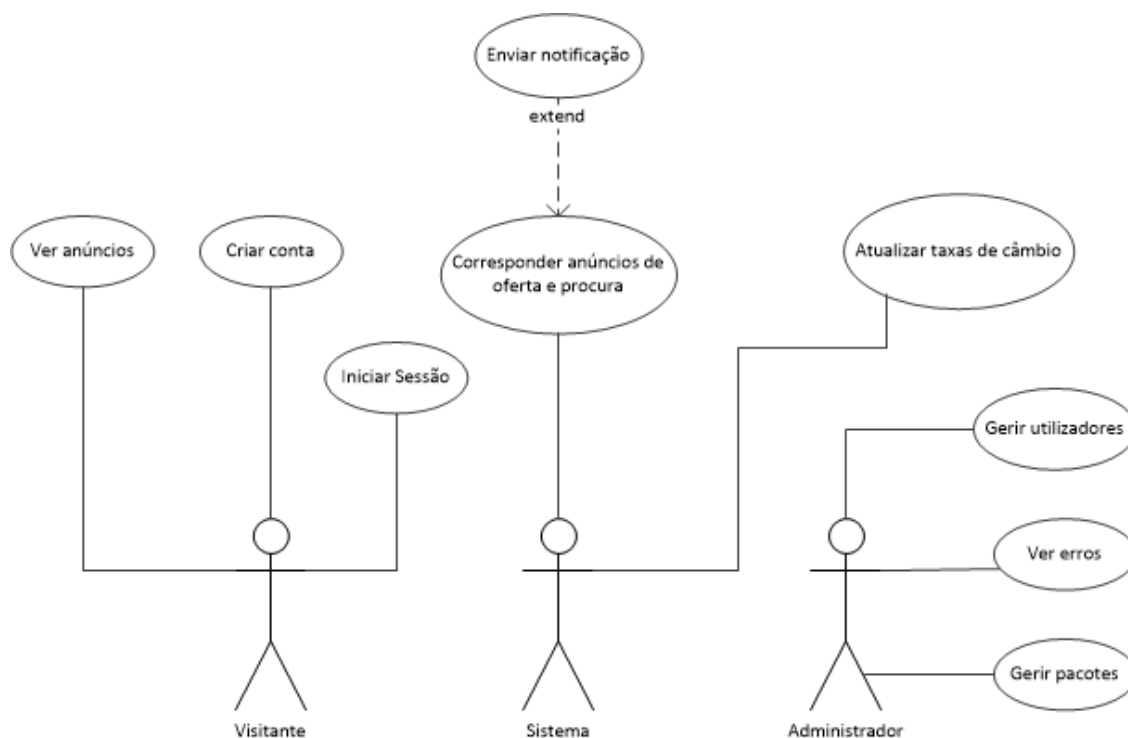


Figura 19 - Diagrama de casos de uso Parte 4

(Fonte: autor)

O Visitante vê os últimos dez anúncios criados por utilizadores de todas as modalidades e poderá registar-se para ver mais anúncios desportivos ligados a uma modalidade.

O Sistema faz corresponder os anúncios desportivos de oferta e de procura que por sua vez envia notificações com informação que seja do interesse do utilizador e atualiza as taxas de câmbio regularmente através de um pedido a um *web service* externo¹⁸.

Para concluir, o Administrador efetua a gestão de todos os utilizadores, observa todos os erros gerados no sistema e gere os pacotes que os utilizadores poderão adquirir.

¹⁸ Web reference: <http://currencyconverter.kowabunga.net/converter.asmx?WSDL>

4.1.3 Diagramas de sequência de sistema

Neste subcapítulo é ilustrado as interações descritas nos casos de uso mencionados anteriormente.

A figura 20 ilustra o diagrama de sequência do caso de uso “Criar conta” do ator Visitante.

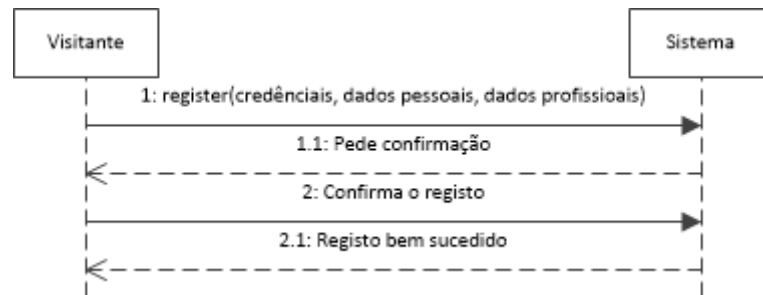


Figura 20 - Diagrama de sequência “Criar conta”

(Fonte: autor)

Neste diagrama ilustra o ator Visitante a efetuar o seu registo preenchendo as credenciais, dados pessoais e dados profissionais manualmente. Este poderá optar por preencher os seus dados pessoais e credenciais de forma automática através do Facebook ou Google.

Na figura 21 está ilustrado o diagrama de sequência referente ao caso de uso “Iniciar Sessão” do ator visitante.

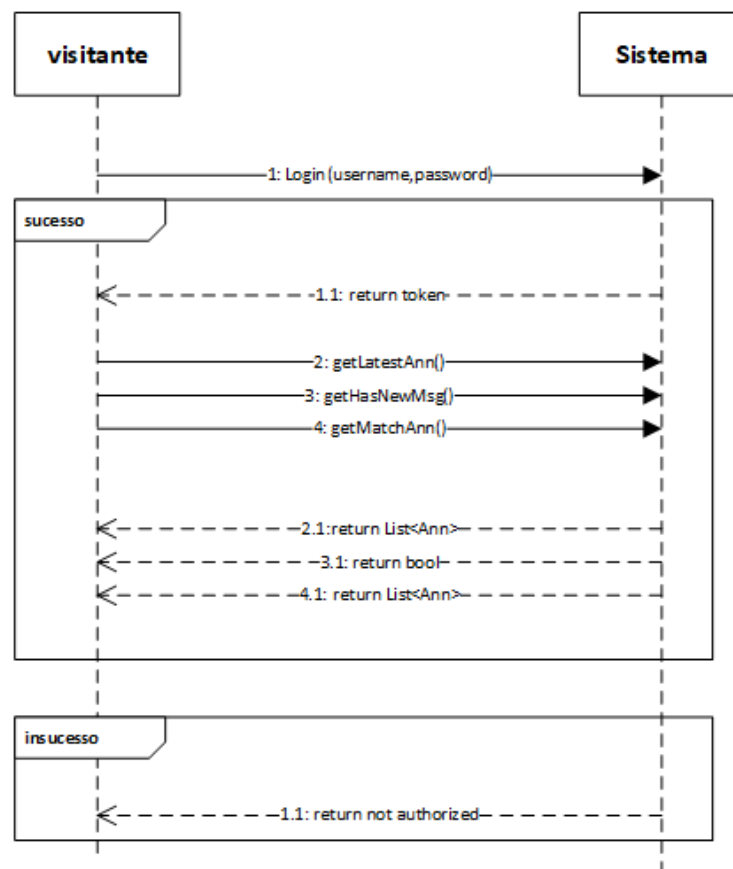


Figura 21 - Diagrama de sequência “Iniciar sessão”

(Fonte: autor)

Após o utilizador inserir as suas credenciais de acesso, o sistema irá validar e em caso de sucesso apresenta uma lista com os últimos anúncios referentes à sua modalidade, verificar se tem novas mensagens e por fim apresenta o número de anúncios que possam ser do seu interesse. Em caso de insucesso é apresentada uma mensagem de credenciais inválidas.

Na figura 22 estão ilustrados vários casos de uso por possuírem o mesmo comportamento entre o Utilizador e Sistema. Os casos de uso são “Gerir anúncios”, “Gerir vídeos” e “Gerir Perfil”.

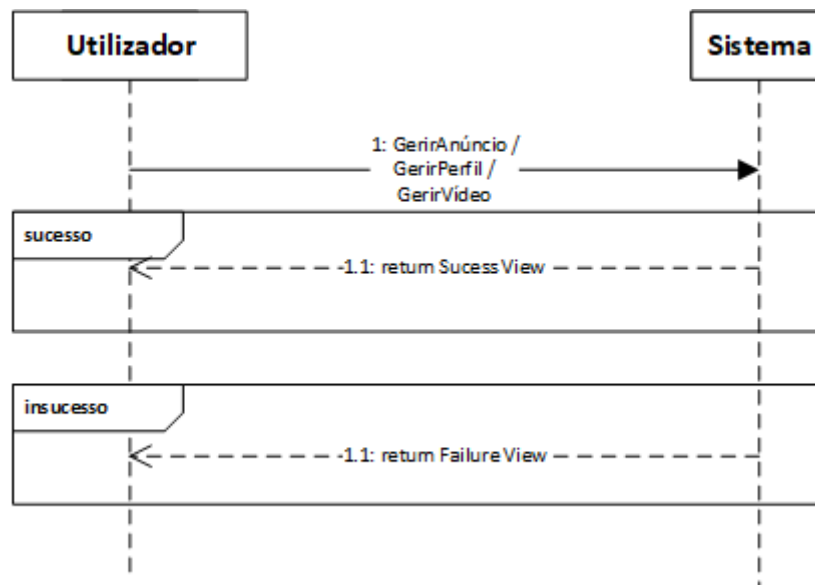


Figura 22 - Diagrama de sequência de gestão de anúncios, vídeos e pacotes

(Fonte: autor)

No diagrama apresentado, o Utilizador comunica com o Sistema para efetuar uma gestão do seu perfil e anúncios. Só o ator Jogador, que herda do ator Utilizador, tem a capacidade de realizar a gestão de vídeos. Entende-se por gestão a inserção, edição e eliminação de um componente.

Na figura 23 está ilustrado o diagrama de sequência referente ao caso de uso “Comprar pacote *premium*” do ator utilizador.

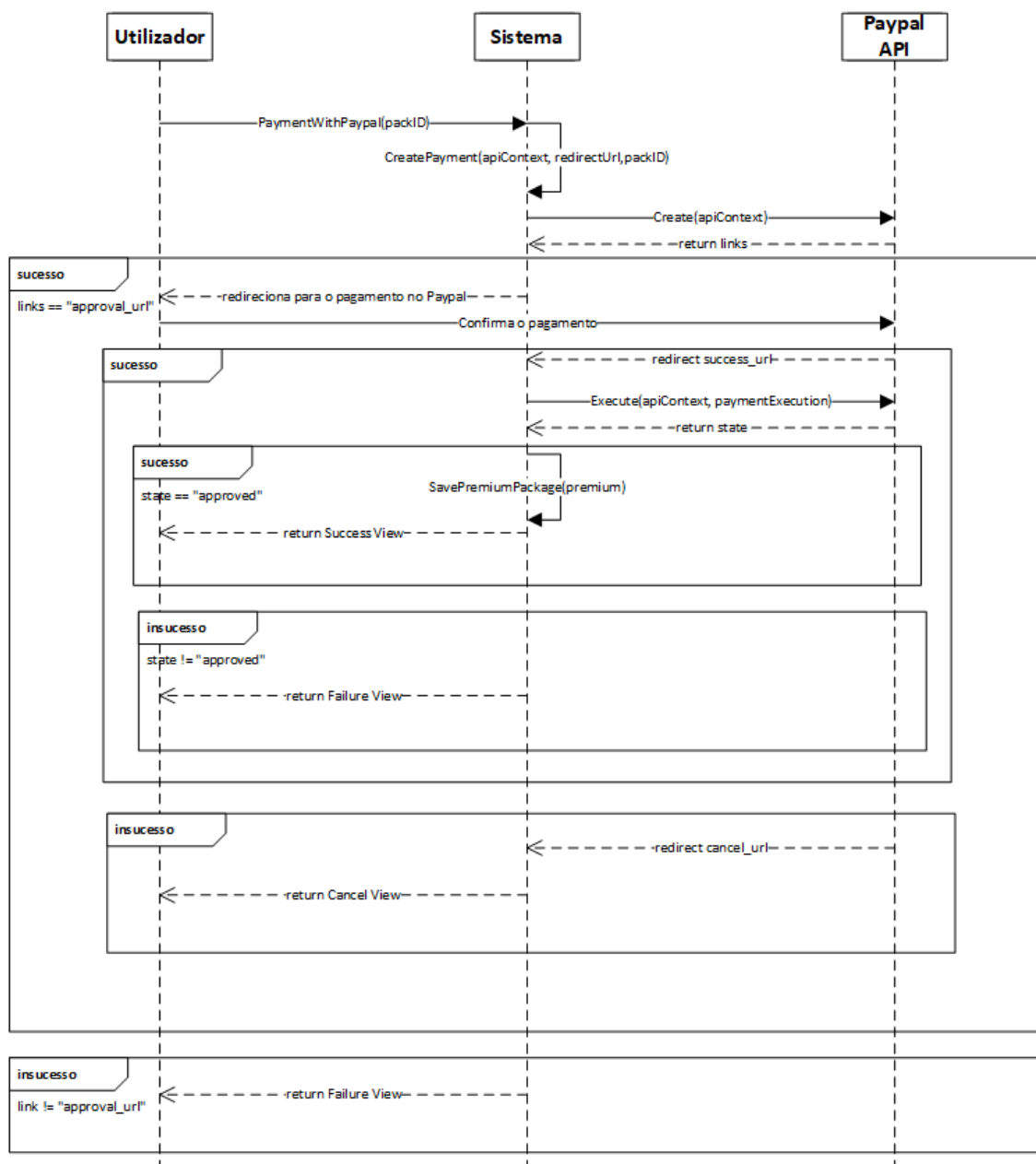


Figura 23 - Diagrama de sequência “Comprar pacote *premium*”

(Fonte: autor)

O Utilizador escolhe um pacote disponível e é direcionado para uma página Paypal, onde este terá de possuir uma conta Paypal, realizar o *login* e confirmar o pagamento. Após a confirmação, o sistema procede à execução do pagamento e em caso de sucesso guarda a informação que o utilizador adquiriu um novo pacote e uma mensagem de sucesso é apresentada. Em caso de existir alguma falha ou cancelamento é apresentada a devida mensagem ao utilizador.

Na figura 24 está ilustrado o diagrama de sequência referente ao caso de uso “Atualizar taxas de câmbio” do ator Sistema.

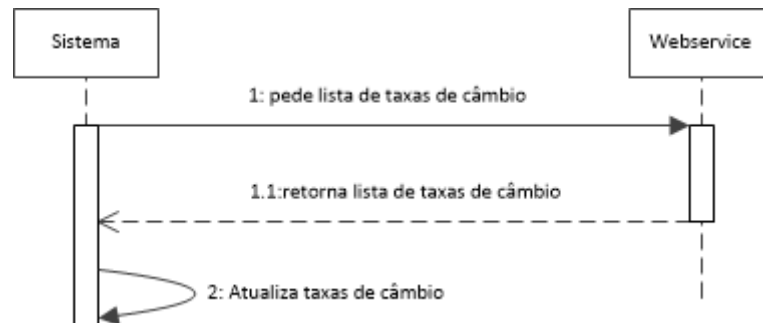


Figura 24 - Diagrama de sequência “Atualizar taxas de câmbio”

(Fonte: autor)

O Sistema envia um pedido a um *web service* externo para retornar uma lista com as taxas de câmbio e posteriormente atualiza-as na sua base de dados. Este processo está configurado para correr diariamente.

Na figura 25 está ilustrado o diagrama de sequência referente ao caso de uso “Gerir pacotes *premium*” do ator Administrador.

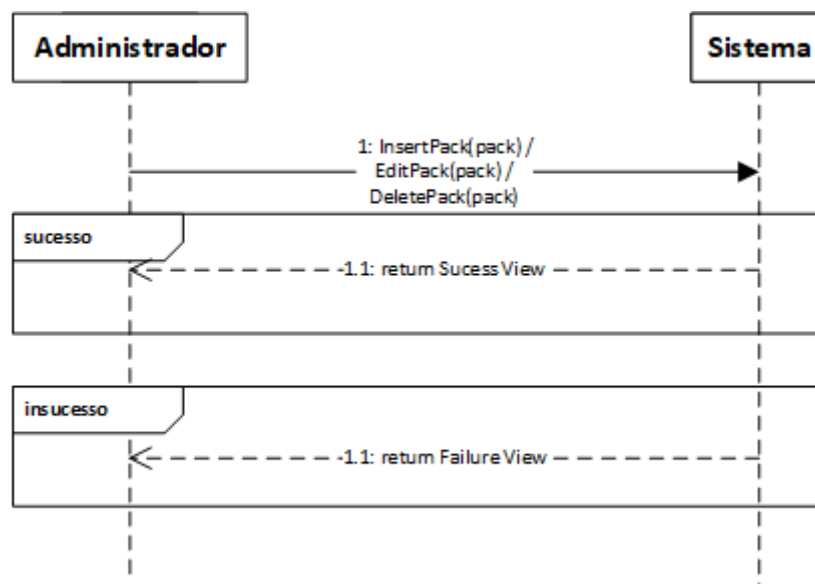


Figura 25 - Diagrama de sequência “Gerir pacotes *premium*”

(Fonte: autor)

O Administrador tem a possibilidade de gerir os pacotes *premium* que a aplicação irá suportar. Numa primeira fase, os pacotes que irão existir são (i) Standard, (ii) Recomendado, (iii) Professional e (IV) Lord (consultar Anexo 2).

4.1.4 Identificação de requisitos não funcionais

"Um requisito não funcional é a descrição de uma propriedade ou característica que um sistema deve apresentar ou um constrangimento que deve ser respeitado" [Wieggers & Beatty, 2013].

Os requisitos não funcionais foram identificados e caracterizados através do padrão FURPS+, sendo eles:

- **Funcionalidade:** é uma categoria com diversas subcategorias que variam de acordo com a aplicação, por exemplo a segurança. A solução tem de implementar mecanismos de segurança para proteger as comunicações e os dados guardados na sua base de dados.
- **Usabilidade:** a solução tem de ser intuitiva para o utilizador e permitir que este interaja de forma rápida e simples com a interface.
- **Desempenho:** a solução tem de correr em servidores com capacidade de processamento suficiente para garantir que os seus serviços respondem aos pedidos dos clientes em tempo útil.

4.1.5 Metodologia

Neste subcapítulo são definidas as metodologias abordadas para o desenvolvimento do projeto.

Foi escolhido o Bitbucket como servidor remoto do repositório Git por questões de simplicidade e pelo autor utilizar uma conta académica, que permite manter o código do projeto num repositório privado.

Para planear, agendar e documentar as tarefas a elaborar no decorrer do projeto foi utilizada a ferramenta JIRA. Esta ferramenta permitiu calcular uma melhor estimativa do tempo a atribuir para a elaboração de cada tarefa, o que facilitou a definição de prioridades entre tarefas.

4.1.6 Modelo relacional

Após a fase de identificação de requisitos funcionais e não funcionais, elaboração dos casos de uso e a definição da arquitetura a implementar segue-se a fase de construção do modelo relacional de dados.

Este modelo é representado através da linguagem UML e é bastante útil para demonstrar:

- A identificação de conceitos (objetos)
- A associação entre conceitos
- Atributos relacionados com os conceitos

Para a implementação da solução foram identificados, numa fase inicial, 24 objetos como ilustra a figura 26.

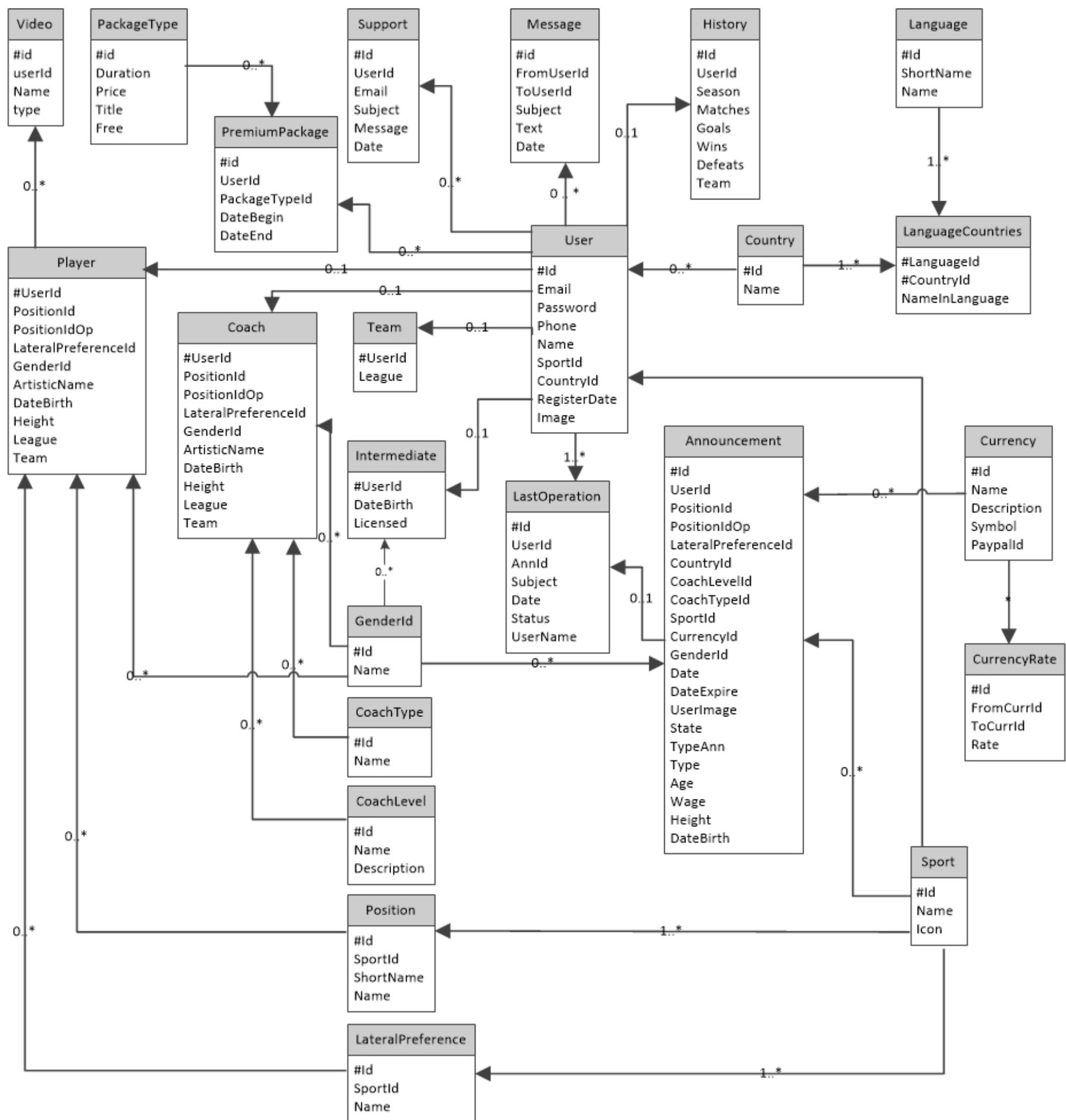


Figura 26 - Modelo relacional de dados

(Fonte: autor)

A entidade User agrupa quatro entidades (Player, Coach, Team e Intermediate) que serão os quatro tipos de utilizadores possíveis a utilizar a solução. Esta entidade está associada às tabelas PremiumPackage, Support, Message, History, Country e Sport de forma a completar o seu perfil, registar no sistema todos os pedidos de ajuda, mensagens para outros utilizadores e os pacotes que pretende adquirir.

A tabela Announcement não está normalizada devido a questões de desempenho da solução, uma vez que esta está envolvida na maior parte das funcionalidades das aplicações *web* e móvel. Assim, os atributos estão armazenados numa só tabela e face à quantidade de pedidos das aplicações é mais vantajoso que não esteja normalizada para aumentar a capacidade de resposta, principalmente sobre os filtros e recomendações de anúncios.

4.2 Implementação do sistema

Nesta subsecção é descrita de uma forma mais detalhada a arquitetura, o processo de autenticação e autorização dos utilizadores, a classificação de anúncios, comunicação entre os diversos componentes que compõe o sistema, dados, serviços, *logs*, idiomas e testes do sistema de recomendação.

4.2.1 Arquitetura

Uma das partes mais importantes do processo de desenvolvimento de *software* é a criação da sua arquitetura, pois representa a forma como os diferentes componentes ou aplicações são organizados e estruturados para formarem um sistema.

Para uma melhor organização do sistema foi utilizado a abordagem Repository Pattern que melhora a facilidade na manutenção e implementação de funcionalidades, elaboração de testes unitários, entre outros. Na figura 27 está representada a arquitetura base de uma abordagem Repository Pattern.

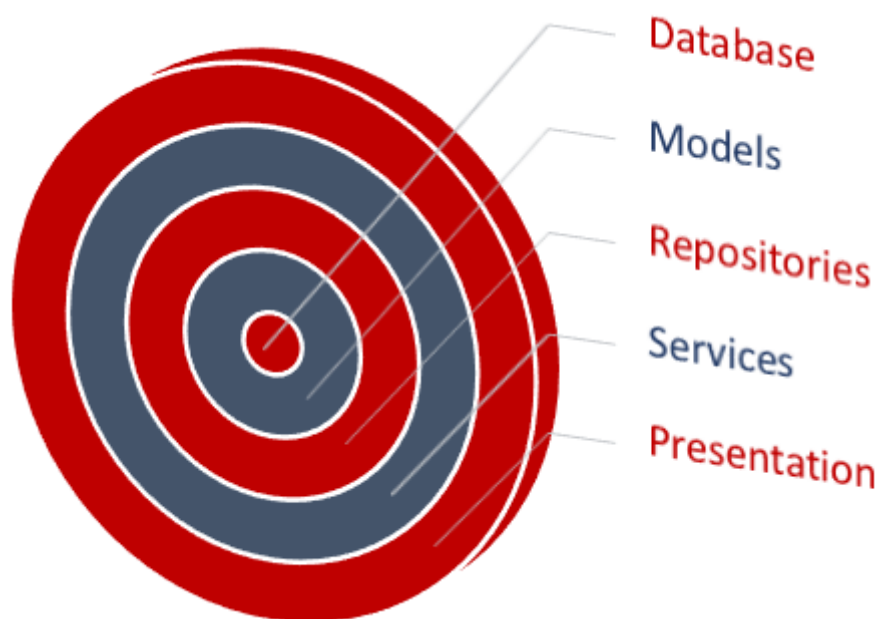


Figura 27 - Arquitetura Repository Pattern

(Fonte: *ASP.NET MVC Solution Architecture – Best Practices* ¹⁹)

A arquitetura da solução *web*, tendo por base a arquitetura Repository Pattern é constituída por 10 componentes:

- **Data:** responsável pela criação de repositórios que comunicarão com a base de dados.
- **Hubs:** responsável por estabelecer uma comunicação entre os utilizadores da aplicação *web* de forma assíncrona na aplicação. O ASP.NET SignalR foi a tecnologia utilizada para facilitar a implementação de uma comunicação em tempo “real” entre utilizadores.
- **Models:** contem o modelo e toda a lógica de negócio.
- **Resources:** componente responsável pela tradução da UI para o idioma que o utilizador pretender. Os idiomas são português, inglês e espanhol.
- **Services:** é dependente dos componentes Data e Models. Este é responsável por todos os serviços da aplicação que exijam a comunicação entre o modelo de negócio e a base de dados.
- **WebApp:** componente responsável pela interação com os utilizadores através do *browser* onde será feita uma abordagem denominada *Responsive Web*

¹⁹ <https://chsakell.com/2015/02/15/asp-net-mvc-solution-architecture-best-practices/> (acedido em 05/02/2018)

Design. Para este componente pretende-se fazer uso de uma arquitetura que seja pouco complexa, escalável, facilmente testável e que se adeque às tecnologias utilizadas, como é o caso de uma arquitetura MVC.

- **MobileApp:** componente responsável pela interação com os utilizadores através de uma aplicação móvel.
- **WebTests:** componente responsável pela realização de testes unitários. Todas as funcionalidades exercidas na aplicação serão testadas com o objetivo de mostrar que cada funcionalidade atende corretamente à sua especificação. Este componente ajuda a encontrar problemas o quanto antes e melhoram a documentação da aplicação. Os testes unitários foram implementados com ajuda de objetos Mock, estes são novas instâncias das classes a utilizar para o teste e simulam o seu comportamento. Assim, todos os objetos Mock estão isolados da aplicação *web* [Microsoft, 2009].
- **WebApi:** componente responsável pela comunicação com a aplicação *web* e móvel ao expor os métodos que estas necessitam.
- **WindowsServices:** componente responsável pela interação com outras plataformas e atualização dos dados recolhidos.

Na figura 28 que se segue é ilustrada a arquitetura lógica do sistema de recomendação.

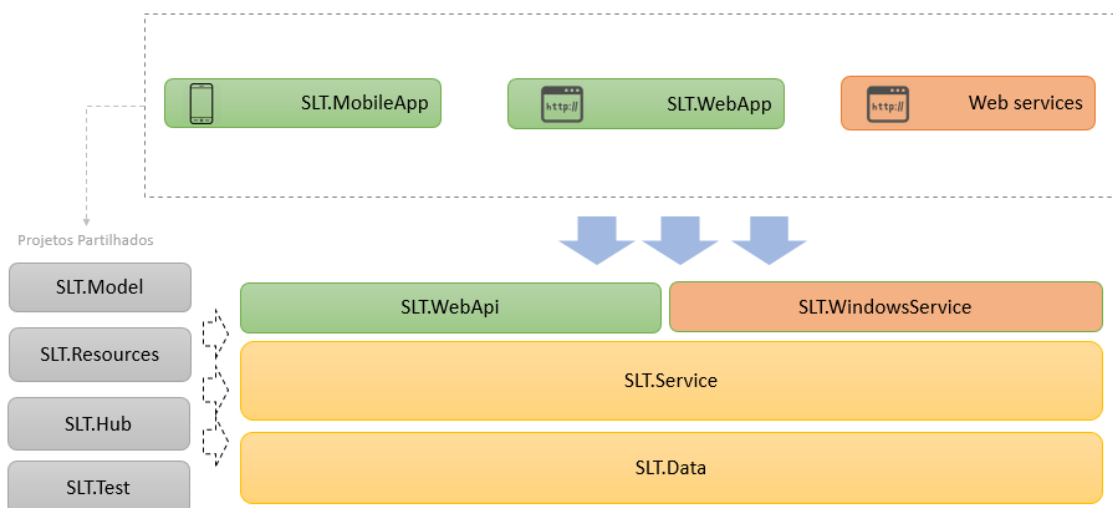


Figura 28 - Arquitetura lógica do sistema

(fonte: autor)

4.2.2 Autenticação e autorização

A tecnologia adotada para implementar a autenticação no sistema de recomendação foi a ASP.NET Identity.

Na fase de implementação para autenticar utilizadores é necessário instalar 3 *packages* (Owin, EntityFramework e SystemWeb) e configurar através do uso das classes IdentityDbContext, UserManager e ClaimsIdentity a conexão à base de dados, gestão das contas dos utilizadores e atribuição de novos roles.

Os roles que compõem o sistema são (I) Admin, (II) Jogador, (III) Equipa, (IV) Agente, (V) Treinador e serão usados para determinar o nível de autoridade que cada utilizador tem para aceder a cada funcionalidade.

Após os utilizadores introduzirem as suas credenciais de acesso e estas serem validadas, o sistema gera um *token* (chave) com uma validade e devolve-o.

Com o acesso ao *token* o cliente (aplicação *web* ou móvel) é capaz de aceder a métodos autenticados ao introduzi-lo no *header* dos pedidos com o parâmetro Authorization. O pedido é intercetado por um filtro onde o *token* é validado e o sistema fica a conhecer o utilizador.

O sistema de recomendação utiliza também um mecanismo de proteção (*captcha*) que evita a introdução de dados em formulários através de *bots*.

4.2.3 Classificação

A correspondência entre anúncios desportivos é uma das peças chave na implementação do sistema, esperando-se que seja também um dos elementos responsáveis pelo seu sucesso. Esta correspondência baseia-se na utilização do algoritmo *K-means* para agrupar os diversos anúncios existentes no sistema, tendo a designação de *clusters*.

Para proceder à sua utilização é crucial realizar a preparação dos dados, constituída pelas fases de seleccionar, limpar e pré-processar os dados. Esta preparação é essencial para produzir modelos válidos e úteis.

Fase de seleção

Nesta fase é definido o conjunto de dados que irá ser processado pelo sistema. Para o anúncio publicado do tipo Oferta irá ser processado um conjunto de dados com anúncios do tipo Procura. Desta forma, o sistema irá corresponder os anúncios do tipo Procura com os do tipo Oferta, e vice-versa.

Os atributos que compõem o conjunto de dados são modalidade, país, idade, posição, nível do treinador, e o tipo de entidade (Jogador ou Treinador).

Fase de limpeza

Nesta fase, foi realizada a limpeza dos dados, ou seja, o tratamento de valores em falta, valores negativos ou nulos.

Fase de pré-processamento

Nesta fase os dados foram convertidos para valores numéricos e categóricos. Os atributos modalidade, país, posição, nível de treinador e entidade são representados através do seu id, enquanto que o atributo idade tem a seguinte categorização:

- Jovem (idades compreendidas entre os 18 e 25 anos)
- Média ((idades compreendidas entre os 26 e 45 anos)
- Avançada (idade superior a 46 anos)

Após esta fase, procedeu-se à normalização do conjunto de dados para a devida utilização do algoritmo. O número de *clusters* (k) definido inicialmente na utilização do algoritmo foi 4.

Após a conclusão desta implementação, foi desenvolvido um *script* na linguagem R, utilizando o RStudio como ambiente de desenvolvimento para realizar alguns testes e analisar o comportamento do algoritmo com o conjunto de dados.

Assim, foi possível determinar que para o conjunto de dados inicial o número de *clusters* ideal é 7, tendo a maior média de coeficiente de Silhouette, com aproximadamente 0,61 (consultar Anexo 6). Esta técnica foi aplicada para inferir a qualidade dos *clusters* gerados pelo algoritmo escolhido.

4.2.4 Comunicação

O projeto Hubs utiliza a tecnologia SignalR (Microsoft.AspNet.SignalR) para manter uma comunicação bidirecional entre o utilizador e o servidor.

Foi implementado uma classe chamada *HomeHub* que possui um método com o nome *sendChatMessage* e este é invocado sempre que um utilizador envia uma mensagem, como exemplifica o seguinte código:

```
var notifications = $.connection.homeHub;  
// Call the Send method on the hub.  
notifications.server.sendChatMessage('@Model.UserIdFrom', '@Model.UserIdTo', msg);
```

Assim, os utilizadores do sistema poderão trocar mensagens através do *chat* de uma forma simples e rápida.

4.2.5 Dados

A criação de repositórios destina-se a criar uma camada de abstração entre a camada de acesso a dados e a camada de lógica de negócio.

Nesta camada de dados foi implementada uma classe principal denominada *SLTEntities*. Esta classe deriva da classe *IdentityDbContext* do EF (Entity Framework) onde são especificadas todas as entidades incluídas no modelo de dados.

Para cada entidade definida é criada uma propriedade *DbSet* que, na terminologia do EF, corresponde a uma tabela na base de dados.

A utilização da abordagem *Code-First* mostrou-se ser a mais indicada para este projeto por se tratar de uma solução construída de raiz e se pretender personalizar as entidades em código e não na base de dados. Por exemplo, a criação dos *roles*, idiomas, épocas desportivas da aplicação, entre outros.

4.2.6 Serviços

O projeto *WindowsServices* é responsável pela comunicação com plataformas externas para obtenção de resultados e implementa um serviço para obter as taxas de câmbio das moedas regularmente e mantê-las atualizadas na base de dados.

O projeto SLT.Service é responsável por toda a parte lógica (regras de negócio) do sistema de recomendação. É da responsabilidade do projeto SLT.WebApi expor os métodos necessários para a comunicação com aplicações externas.

4.2.7 Log

A tecnologia de *logs* utilizada no sistema de recomendação para identificar falhas de acesso à base de dados e exceções não tratadas foi o Elmah. É uma ferramenta *open source* para armazenar todos os erros ocorridos em aplicações .NET e ajuda a identificar falhas no sistema.

Foi adicionada uma tabela, chamada PremiumPackageLog, na base de dados para registar toda a informação sobre a compra de pacotes *premium*. Esta informação será útil para conhecer quais os pacotes que terão mais procura, criar perfis de utilizadores face a cada compra realizada e em caso de existir algum problema com um utilizador ser possível consultar a informação detalhada como o preço, duração e tipo de pacote da sua compra.

4.2.8 Idioma

O projeto Resources contém todos os idiomas que o sistema suporta. Atualmente é composto por três (português, inglês e espanhol).

4.2.9 Testes

A utilização de testes de *software*, quer sejam eles unitários, de integração, aceitação ou outros, permitem que sejam detetados defeitos no sistema implementado.

O projeto WebTests contém testes unitários para a avaliação de funções básicas do sistema e testes de integração para garantir todas as comunicações entre os projetos.

A sua utilização é particularmente útil na introdução de novas funcionalidades de *software*, mas também na alteração de funcionalidades já existentes. O aparecimento de erros em funcionalidades já existentes, nas quais os testes anteriormente passavam, poderá indicar a introdução de novos defeitos.

4.3 Implementação da aplicação web

Nesta subsecção é descrita de uma forma mais detalhada a arquitetura e interface da aplicação *web*.

A aplicação foi implementada na linguagem de programação C# com recurso à *framework* ASP.NET MVC (versão 4.5). Esta aplicação utiliza um *view engine* chamado Razor para simplificar a construção de páginas.

Com a utilização do Razor é possível utilizar a linguagem C# em conjunto com o HTML para facilitar a implementação das páginas da aplicação.

4.3.1 Arquitetura

A arquitetura utilizada nesta aplicação foi a MVC, como referido anteriormente.

Este padrão tem como objetivo dividir a aplicação em três componentes e para cada um atribui as respetivas responsabilidades promovendo um baixo acoplamento e alta coesão, tornando o sistema escalável. Os três componentes são:

- **Model** é responsável pelas regras de negócio e domínio do problema, ou seja, a leitura e escrita de dados e as respetivas validações.
- **View** fornece ao utilizador as informações à cerca do modelo, ou seja, é responsável pela interface da aplicação.
- **Controller** é o responsável pelas interações que utilizador tenha com a UI.

4.3.2 Interface

A ferramenta utilizada para o desenvolvimento da interface da aplicação *web* foi a Bootstrap. Esta, tem se tornado uma das ferramentas mais utilizadas ao longo dos anos na implementação de interfaces para aplicações *web* devido às suas características.

O autor identifica como principais vantagens desta ferramenta (I) a grande biblioteca de componentes (ícones, painéis, etc), (II) documentação, (III) fácil implementação de aplicações responsivas e (IV) a reutilização de código.

Para manipular todos os eventos da interface recorreu-se a uma biblioteca JavaScript chamada JQuery. Esta biblioteca desempenha um papel fundamental na aplicação

porque é responsável por manipular o HTML/DOM/CSS, efeitos, animações e garantir a compatibilidade entre *browsers*.

4.4 Implementação da aplicação móvel

Nesta subsecção é descrita de uma forma mais detalhada a arquitetura, interface da solução móvel e posteriormente os detalhes de implementação para cada sistema operativo (Android e iOS).

Com o aumento do consumo de aplicações móveis e do *e-commerce* as aplicações começaram a interagir mais com os utilizadores promovendo produtos ou serviços que estes possam estar interessados em adquirir.

A aplicação foi implementada na tecnologia React Native (versão 0.50.1) que permite a criação em simultâneo de aplicações Android e iOS. O RN compila o código desenvolvido em JavaScript e faz uso dos mesmos componentes nativos de cada sistema (proporcionando assim uma experiência mais nativa aos utilizadores).

4.4.1 Arquitetura

Para a implementação da aplicação móvel foi definida a seguinte arquitetura:

- **Android:** camada onde persiste o código nativo para a plataforma Android.
- **iOS:** camada onde persiste o código nativo para a plataforma iOS.
- **Source:** camada onde persiste o código comum às duas plataformas.
 - **Css:** componente responsável pelos estilos aplicados na aplicação móvel.
 - **Helpers:** componente que é responsável pelos métodos auxiliares a várias classes da aplicação.
 - **Images:** componente onde persiste todas as imagens incorporadas na aplicação.
 - **Resources:** componente responsável pela tradução dos idiomas da aplicação.
 - **Views:** componente responsável pela interação com o utilizador e o servidor.

4.4.2 Interface

A interface da aplicação móvel foi desenvolvida através de componentes de RN juntamente com CSS para a aplicação de estilos. É realçada a vantagem de utilizar componentes já existentes na aplicação desenvolvidos pela comunidade.

Para um melhor detalhe de quais os componentes e respetivas versões utilizadas na implementação da aplicação é necessário consultar o Anexo 4.

4.4.3 Android

No desenvolvimento da aplicação móvel para Android foi utilizado um gestor de dispositivos do Android Studio para criar e configurar um dispositivo virtual (AVD) para correr num emulador. O emulador configurado simula um dispositivo com o sistema operativo Android 6.0 (API 23) de forma a ser possível correr e testar a aplicação.

Para configurar as notificações em sistemas Android foi utilizado o FireBase Cloud Messaging (FCM). Após a criação da aplicação no FCM²⁰ é necessário transferir e guardar o ficheiro google-services.json para a raiz do módulo da aplicação Android e efetuar algumas configurações nos ficheiros build.gradle e androidmanifest.xml para incluir o serviço Google.

Numa primeira fase as notificações foram enviadas com sucesso para o emulador e dispositivo físico através do *browser* pelo FCM. A fase seguinte consistiu em enviar as notificações através do portal do Azure²¹.

Após concluir a fase anterior com sucesso, surgiu a ideia de enviar notificações *push* diretamente a partir de uma aplicação utilizando a API REST do Hub Notification do Azure. A ideia surgiu já numa fase final do projeto pelo que não foi possível concluí-la.

²⁰ Console.firebse.com (acedido em 16/05/2018)

²¹ <https://docs.microsoft.com/pt-pt/azure/notification-hubs/notification-hubs-android-push-notification-google-fcm-get-started> (acedido em 16/05/2018)

4.4.4 iOS

No desenvolvimento da aplicação móvel para iOS foi utilizado o emulador do Xcode (IDE de desenvolvimento da Apple) para testar a aplicação móvel em diferentes dispositivos.

Para as notificações *push* funcionarem corretamente em dispositivos iOS é necessário proceder a algumas configurações específicas nesta aplicação, tal como se sucedeu na aplicação Android. Para tal, é necessário configurar um cliente iOS²² e configurar o APNS (Apple Push Notification Service) com o FCM.

Numa primeira fase as notificações foram enviadas com sucesso para o dispositivo físico através do *browser* pelo FCM. Note-se que o emulador não suporta notificações.

4.5 Base de dados

Existem várias opções para a implementação de uma base de dados, tal como SQL Server e Oracle. Para o sistema foi criada e armazenada uma base de dados relacional, em SQL Server pela facilidade do seu crescimento, conhecimento prévio da linguagem e melhor integração com a *framework* ASP.NET MVC.

O *software* para gestão da base de dados utilizado foi o SQL Management Studio da Microsoft

²² <https://firebase.google.com/docs/cloud-messaging/ios/client> (acedido em 17/05/2018)

5 Avaliação da solução

Neste capítulo, pretende-se descrever de forma sucinta, para o problema e objetivos definidos, quais foram as grandezas, hipóteses, metodologia de avaliação e resultados que representaram a base para avaliação da solução implementada.

5.1 Grandezas e Hipóteses

De forma a validar se a solução apresentada é ou não uma mais-valia, relativamente às soluções existentes no mercado, são definidas como grandezas a usabilidade da solução e a satisfação dos seus utilizadores.

Estas grandezas são indicadores para avaliar o grau de usabilidade que cada cliente irá ter com as aplicações implementadas e perceber a sua satisfação.

As hipóteses a serem testadas são (i) perceber se o utilizador utiliza as aplicações implementadas e (ii) qual o grau de satisfação dos utilizadores através dos seguintes indicadores: utilidade, desempenho, facilidade e funcionalidade das aplicações implementadas. Assim, é possível avaliar a qualidade da solução e o valor para o cliente.

5.2 Metodologia de avaliação

A avaliação da solução apresentada foi efetuada com recurso a um inquérito de satisfação e de usabilidade (consultar Anexo 5). É importante apurar o quão satisfeito está o utilizador. Caso esteja insatisfeito, é necessário perceber o motivo.

As questões do inquérito sobre usabilidade e satisfação com classificação de 1 a 5, representada na tabela 14, são:

- 1 - A utilização da solução é intuitiva e simples?
- 2 - A aplicação *web* responde rapidamente quando aplicados os filtros?
- 3 - A aplicação móvel responde rapidamente quando aplicados os filtros?
- 4 - A aplicação móvel representa uma mais-valia?
- 5 - Os anúncios notificados vão de encontro às suas necessidades?
- 6 - Qual o desempenho da aplicação *web*?
- 7 - Qual o desempenho da aplicação móvel?

- 8 - Este sistema poderá ser útil para a sua carreira profissional?

Tabela 14 - Escala de classificação utilizada no inquérito

Escala	Descrição
1	Muito insatisfatório
2	Insatisfatório
3	Sem opinião
4	Satisfatório
5	Muito satisfatório

5.3 Resultados do inquérito

A qualidade dos resultados é um dos principais fatores na análise de qualquer solução.

Os resultados do inquérito estão representados na tabela 15 para uma melhor perceção por parte do leitor e foram obtidas 32 respostas.

Tabela 15 - Resultado do inquérito de satisfação e de usabilidade

Pergunta	Classificação				
	1	2	3	4	5
1	0%	0%	0%	16%	84%
2	0%	0%	0%	0%	100%
3	0%	0%	25%	9%	66%
4	0%	0%	25%	0%	75%
5	0%	0%	47%	0%	53%
6	0%	0%	0%	0%	100%
7	0%	0%	25%	0%	75%
8	0%	0%	16%	50%	34%

Com base, nos resultados obtidos é possível verificar que nas perguntas (3, 4, 7) que envolvem a aplicação móvel, existem 8 respostas sem opinião. Provavelmente, estes utilizadores ainda não instalaram e usufruíram da aplicação móvel.

As respostas à pergunta 5 estão bastante divididas entre a classificação 3 e 5. É normal nesta fase obter estas classificações pelo sistema ser recente e existirem utilizadores que não obtiveram qualquer recomendação de anúncios desportivos que correspondam com o seu perfil.

As perguntas 2 e 6, relacionadas com o desempenho da aplicação *web*, obtiveram a mesma resposta (muito satisfatório) por todos os utilizadores.

Em relação às respostas da pergunta 8, conclui-se que há utilizadores bastante otimistas e outros que aguardam resultados concretos para se poderem pronunciar.

De acordo, com os indicadores apresentados anteriormente é possível concluir que:

- **Utilidade:** de um modo geral, os resultados são bastante satisfatórios sobre a utilidade da solução.
- **Desempenho:** todos os resultados (100%) sobre o desempenho da aplicação *web* são muito satisfatórios. Em relação ao desempenho da aplicação móvel existem cerca de 8 resultados sem opinião.
- **Facilidade:** os resultados obtidos são muito satisfatórios pela facilidade com que o utilizador interage com a solução.
- **Funcionalidade:** com base nos resultados obtidos, as funcionalidades chave da solução estão a funcionar corretamente e com um bom desempenho.

5.4 Conclusão

Com base em todas as respostas obtidas, apesar de terem sido efetuadas por uma amostra relativamente pequena, pode-se concluir que os resultados foram positivos apresentando um grau de satisfação e usabilidade elevado por parte dos inquiridos.

É possível concluir que a modalidade que abrange um maior número de utilizadores é o futebol (63%) com um total de 20 inquiridos, seguido de basquetebol (22%) com 7, andebol (9%) com 3 e por último o voleibol (6%) com 2.

Sendo este um sistema global, verificou-se que o sistema atingiu 6 países diferentes como a Alemanha (16%) com 5 inquiridos, Argentina (6%) com 2 inquiridos, Azerbaijão (9%) com 3 inquiridos, Brasil (19%) com 6 inquiridos, Índia (9%) com 3 inquiridos e por último Portugal (41%) com 13 inquiridos.

Seria interessante numa fase posterior efetuar novamente um inquérito para analisar e avaliar se com uma amostra maior os resultados obtidos seriam diferentes.

6 Conclusão

O último capítulo da presente dissertação comporta o balanço do projeto implementado. Aqui são expostos os objetivos alcançados, as dificuldades encontradas ao longo de todo o percurso, assim como possíveis trabalhos decorrentes da ideia central deste projeto.

6.1 Principais conclusões

De acordo com os objetivos principais desta dissertação, foi realizada uma investigação sobre sistemas de recomendação no capítulo 2. Neste capítulo, são apresentados os primeiros sistemas a atuar no mercado, alguns sistemas existentes atualmente na área do desporto, sistemas na área de Recursos Humanos, semelhantes ao sistema implementado e o estudo de possíveis tecnologias para a implementação da solução.

No capítulo 3, é realizada uma análise de valor onde é definida a proposta de valor e o valor percebido, de forma a perceber as vantagens de implementação da solução para os clientes. Através do método AHP, foi realizada uma comparação entre tecnologias e concluiu-se qual a melhor a utilizar em cada aplicação face aos quatro critérios definidos.

No capítulo 4 é apresentada toda a análise e implementação da solução e concluiu-se que as tecnologias escolhidas para o desenvolvimento da solução foram uma boa escolha porque permitiram acelerar o processo de implementação da solução final através de bibliotecas já existentes em C# para a aplicação *web* e de componentes em JS para as aplicações móveis. Estas tecnologias têm como principal vantagem a sua vasta documentação, de fácil consulta.

Por fim, no capítulo de Avaliação da solução é apresentado o inquérito de satisfação e de usabilidade onde os clientes da solução atribuíram uma classificação e concluiu-se que os resultados foram positivos apresentando um grau de satisfação e usabilidade elevado por parte dos inquiridos.

6.2 Objetivos alcançados

O principal objetivo deste projeto consistia na implementação de um sistema de recomendação na área do desporto, que permitisse criar uma correspondência entre a oferta e a procura de anúncios desportivos com base no perfil de cada utilizador que foi alcançado com sucesso, existindo uma satisfação manifestada por parte de todos os intervenientes.

No decorrer do projeto, surgiram outros objetivos importantes como:

- Implementar notificações *push* nas aplicações móveis, que ficaram implementadas apenas em ambiente de desenvolvimento devido à inexistência de alguns recursos.
- O utilizador adquirir pacotes *premium* através de um pagamento efetuado por uma conta Paypal, que foi concluído com sucesso.

6.3 Limitações e Trabalho futuro

Como principais limitações existentes no decorrer do projeto foram identificadas o tempo de implementação do projeto, o acesso ao servidor para configurar o ambiente de produção e uma fraca amostra de dados no inquérito realizado aos utilizadores do sistema.

Entende-se por trabalho futuro, encontrar meios para solucionar os problemas encontrados durante a implementação do projeto.

De forma a dar continuidade ao projeto e a torná-lo mais robusto, são identificados os seguintes pontos a melhorar e a implementar no futuro:

- Implementar uma nova aplicação (*Backoffice*) para dar suporte à gestão do sistema de recomendação e para envio de notificações aos utilizadores. Este objetivo não foi alcançado devido ao tempo, mas existe uma área na aplicação *web* destinada para o Administrador, onde este pode consultar todos os erros ocorridos e gerir os componentes mais importantes do sistema.
- Criação de padrões nos dados através da utilização de algoritmos de *data mining* para a criação de perfis de utilizadores.
- Possibilidade de expansão do sistema para outras modalidades e abranger novas entidades.

- Obter um maior número de utilizadores para efetuar uma reavaliação do sistema implementado para possíveis novas conclusões.

O sistema de recomendação implementado poderá ser aplicado também a outras áreas, como a área de recursos humanos, uma vez que esta recomenda candidatos com base no seu perfil face a uma determinada oferta.

Referências

Alinone, A. (2007). *Comet and Push Technology*.

Alinone, A. (2011). *10 Years of Push Technology, Comet, and WebSockets*.

Balabanovic, M. & Shoham, Y. (1997). *Fab: content-based, collaborative recommendation*. (Special Section: Recommender Systems).

Beaulieu, A. (2009). *Learning SQL, 2nd Edition*.

Bookkrajang, J. (2017) *Recommendation System*.

Casaca, C. et al (2010). *Recommender Systems for Human Resources Task Assignment*.

Comité Olímpico de Portugal (2012). *Desporto, crescimento económico e emprego*.

Eisenman, B. (2016). *Learning React Native – Building Native Mobile Apps with Javascript*.

Federação Portuguesa de Futebol (2009). *Regulamento relativo ao Estatuto e Transferências de Jogadores, artigo nº19*.
(url: http://www.afl.pt/_docs/190_reg_estat_transf_jog_FIFA.pdf)

Fisher, B. & Chen, J. (2014). *Flux: An Application Architecture for React*.
(url: <https://reactjs.org/blog/2014/05/06/flux.html>)

Francesco Ricci, Lior Rokach and Bracha Shapira (2011). *Introduction to Recommender Systems Handbook*.

Goldberg, D. et al. (1992). *Using collaborative filtering to weave an information Tapestry*.

Guthrie, S. (2016). *Microsoft to acquire Xamarin and empower more developers to build apps on any device*.
(url: <https://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/>)

Holdener, A. (2008). *Ajax: The Definitive Guide, 1st edition*. Beijing; Cambridge MA: O'Reilly Media.

Huang, A. (2008). *Similarity Measures for Text Document Clustering*.

Jannach, D. et al. (2011). *Recommender Systems: An Introduction*.

Jornal Económico (2016). *Harpoon.jobs – o match perfeito entre talentos e empresas*.
(url: <http://www.jornaleconomico.sapo.pt/noticias/harpoon-jobs-match-perfeito-talentos-empresas-69377>)

Koen, P. et al. (2001). *Providing Clarity and a Common Language To the «Fuzzy Front End.» Research Technology Management*.

Koen, P. et al. (2002). *Fuzzy Front End: Effective Methods, Tools, and Techniques*.

Lawrence, M. (2015). *Techniques of value analysis and engineering*.

Lemire, D & Maclachlan, A. (2005). *Slope One Predictors for Online Rating-Based Collaborative Filtering*.

Microsoft (2009). *Unit Testing with Mock Objects*.
(url: <https://msdn.microsoft.com/en-us/pt/library/ff650441.aspx>)

Mojica, J. (2003). *C# Web Development with ASP.NET*.

Osterwalder, A. & Pigneur, Y. (2011). *Business Model Generation: Inovação em Modelos de Negócios*. Rio de Janeiro, RJ: Alta Books.

Pazzani, M. & Billsus, D. (2007). *Content-Based Recommendation Systems*.

Pimentel, D. (2014). *Web Solution to support real time systems*.
(url: <http://hdl.handle.net/10773/14712>)

Rodrigues F. (2016). *Descoberta de conhecimento – Clustering*.

Souza, R. (2014). *Sistemas de Recomendação*.
(url: https://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao/)

Tan, P. et al. (2006). *Introduction to Data Mining, chap: 8*.

Ulaga, W. & Eggert, A. (2006). *Relationship value and relationship quality: broadening the nomological network of business-to-business relationships*. *European Journal of Marketing*.

Wieggers, K. & Beatty, J. (2013). *Software Requirements 3rd edition*.

Williamson, K. (2015). *Introdução ao AngularJS: Um guia para o desenvolvimento com o AngularJS*. Novatec Editora. Ltda.

Anexos

1. Modalidades do sistema

Andebol, Basquetebol, Ciclismo, Cricket, Futebol, Futsal, Golf, Hoquei, Rugby, Tennis e Voleibol.

2. Tipos de pacotes *premium*

Standard, Recomendado, Profissional e Lord.

3. Moedas suportadas pelo sistema

EUR, USD, GBP, INR, AUD, JPY, CHF, HUF, CAD, BRL, AOA, TRY e CNY.

4. Componentes utilizados na aplicação móvel

```
"dependencies": {  
  
  "base-64": "^0.1.0",  
  
  "decode-html": "^2.0.0",  
  
  "he": "^1.1.1",  
  
  "html-entities": "^1.2.1",  
  
  "moment": "^2.20.1",  
  
  "react": "^16.0.0",  
  
  "react-collapsible": "^2.0.3",  
  
  "react-dom": "^16.1.1",  
  
  "react-native": "^0.50.1",  
  
  "react-native-calendar-select": "^0.1.2",  
  
  "react-native-calendars": "^1.17.2",  
  
  "react-native-carousel": "^0.11.0",  
  
  "react-native-chooser": "^1.7.0",  
  
  "react-native-clean-project": "^1.0.6",  
}
```

```

"react-native-collapsible": "^0.9.0",
"react-native-elements": "^0.17.0",
"react-native-html-to-text": "0.0.5",
"react-native-localization": "^0.2.1",
"react-native-material-dropdown": "^0.9.0",
"react-native-modal-dropdown": "^0.5.0",
"react-native-simple-store": "^1.3.0",
"react-native-vector-icons": "^4.4.0",
"react-native-view-more-text": "^1.3.1",
"react-navigation": "^1.5.2",
"xmldom": "^0.1.27"
}

```

5. Resultados do inquérito de Satisfação e Usabilidade

Pergunta 1

A utilização da solução é intuitiva e simples?



Pergunta 2

A aplicação web responde rapidamente quando aplicados os filtros?



Pergunta 3

A aplicação móvel responde rapidamente quando aplicados os filtros?



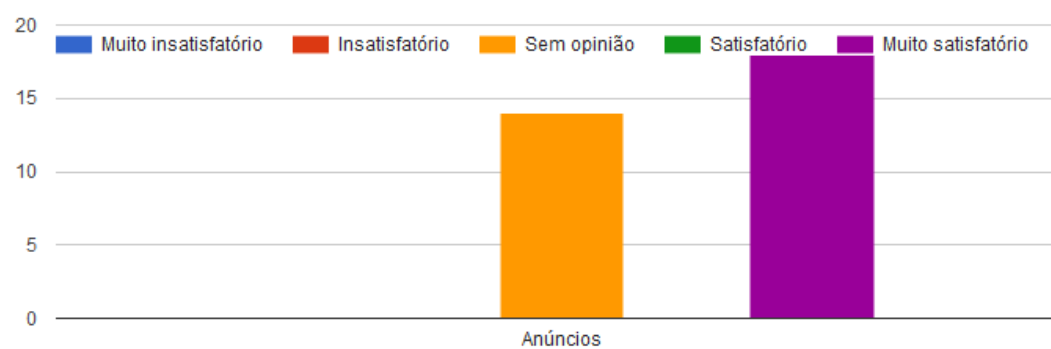
Pergunta 4

A aplicação móvel representa uma mais-valia?



Pergunta 5

Os anúncios notificados vão de encontro às suas necessidades?



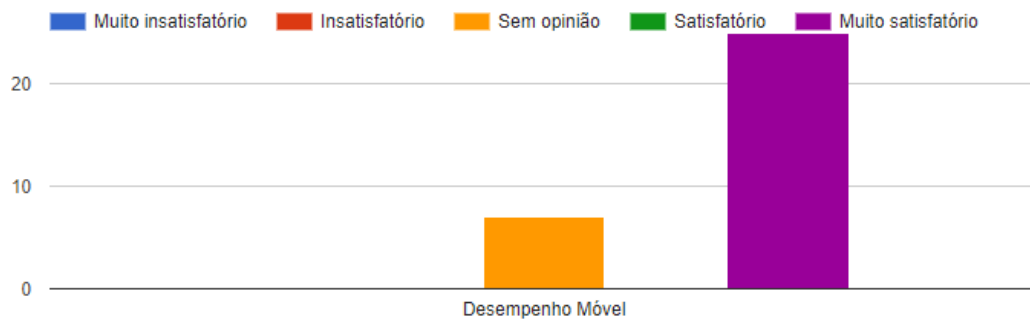
Pergunta 6

Qual o desempenho da aplicação web?



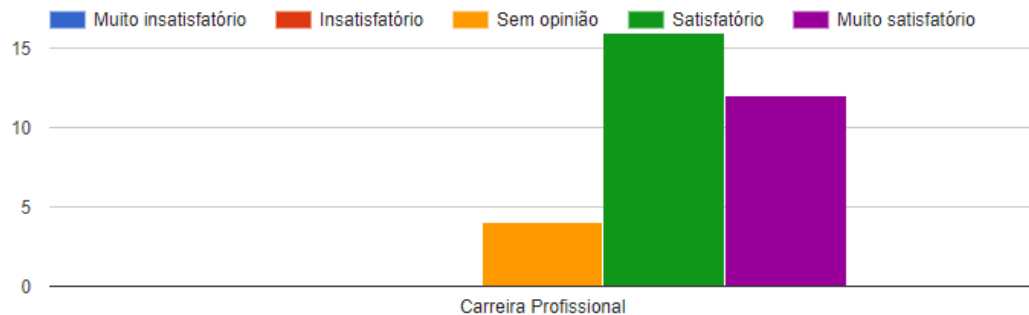
Pergunta 7

Qual o desempenho da aplicação móvel?



Pergunta 8

Este sistema poderá ser útil para a sua carreira profissional?



6. Média do coeficiente de Silhouette para cada k

```
> avgSilhouetteWidth
k averagesilhouette
1 3      0.4726533
2 4      0.4942436
3 5      0.5390556
4 6      0.5718472
5 7      0.6060813
6 8      0.5963413
```

7. Quantidade de perguntas de cada tecnologia web na comunidade StackOverflow e Github

- Na comunidade StackOverflow existem cerca de 263.000 perguntas relacionadas com a tecnologia .NET e na comunidade GitHub mais de dois milhões.
- Na comunidade StackOverflow existem cerca de 250.000 perguntas relacionadas com a tecnologia AngularJS e na comunidade GitHub cerca de 60.000.
- Na comunidade StackOverflow existem cerca de 80.000 perguntas relacionadas com a tecnologia Laravel e na comunidade GitHub cerca de 100.000.

8. Quantidade de perguntas de cada tecnologia móvel na comunidade stackoverflow e github

- Na comunidade StackOverflow existem aproximadamente 74.000 perguntas relacionadas com a tecnologia Xamarin e na comunidade GitHub cerca de 35.000.
- Na comunidade StackOverflow existem aproximadamente 57.000 perguntas relacionadas com a tecnologia React Native e na comunidade GitHub é superior a 100.000.